

III. Papers Filed by Appellants (Relevant Prosecution History)

Request for Continued Examination dated July 23, 2007

Final Office Action dated January 23, 2008

Response to Final Office Action filed July 21, 2008

CERTIFICATE OF ELECTRONIC FILING UNDER 37 CFR 1.8
I hereby certify that this correspondence is being electronically
filed with the U.S. Patent & Trademark Office on 23 July 2007.

/Brianna Dahlberg/
Brianna Dahlberg

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Bart Alan MELTZER et al.

Application No.: 09/173,858

Confirmation No.: 4734

Filed: 16 October 1998

Title: **Documents for Commerce in Trading
Partner Networks and Interface
Definitions Based on the Documents**

Group Art Unit: 2178

Examiner: HUYNH, Cong Lac T.

CUSTOMER NO.: 22470

Mail Stop RCE
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REQUEST FOR CONTINUED EXAMINATION FOLLOWING APPEAL

Madame:

Following the Decision On Request for Rehearing dated May 21, 2007,
Applicants request entry of the following new evidence and consideration of the
following remarks. A Request for Continued Examination (RCE) accompanies this
paper.

The Claims and their current status are reflected beginning on page 2.

Remarks begin on page 6.

The Evidence Appendix begins on page 36.

CLAIMS APPENDIX

1. (Previously presented) An interface for transactions among nodes in a network including a plurality of nodes which execute processes involved in the transactions, the interface being stored in a computer readable medium, comprising:

a machine readable specification of an interface to transaction processes stored in memory accessible by at least one node in the network, including interpretation information providing a definition of an input document, and a definition of an output document, the definitions of the input and output documents comprising respective descriptions of sets of storage units and logical structures for the sets of storage units.
2. (Original) The interface of claim 1, wherein the interpretation information includes data type specifications for at least one logical structure in the definitions of the input and output documents.
3. (Original) The interface of claim 1, wherein the interpretation information includes at least one data structure mapping predefined sets of storage units for a particular logical structure in the definitions of the input and output documents, to respective entries in a list.
4. (Original) The interface of claim 1, including a repository in memory accessible by at least one node in the network storing a library of logical structures, and interpretation information for logic structures.
5. (Original) The interface of claim 1, wherein the machine readable specification includes a document compliant with a definition of an interface document including logical structures for storing an identifier of a particular transaction, and at least one of definitions and references to definitions of input and output documents for the particular transaction.
6. (Original) The interface of claim 1, wherein the machine readable specification includes a document compliant with a definition of an interface document including logical structures for storing an identifier of the interface, and for storing at least one of specifications and references to specifications of a set of one or more transactions supported by the interface.

7. (Original) The interface of claim 6, wherein the machine readable specification includes a reference to a specification of a particular transaction, and the specification of the particular transaction includes a document including logical structures for storing at least one of definitions and references to definitions of input and output documents for the particular transaction.
8. (Original) The interface of claim 1, wherein the storage units comprise parsed data.
9. (Original) The interface of claim 8, wherein the parsed data in at least one of the input and output documents comprises:
 - character data encoding text characters in the one of the input and output documents, and
 - markup data identifying sets of storage units according to the logical structure of the one of the input and output documents.
10. (Original) The interface of claim 9, wherein at least one of the sets of storage units encodes a plurality of text characters providing a natural language word.
11. (Original) The interface of claim 8, wherein the interpretation information for at least one of the sets of storage units identified by a particular logical structure of at least one of the input and output documents, encodes respective definitions for sets of parsed characters.
12. (Original) The interface of claim 8, wherein the storage units comprise unparsed data.
13. (Original) The interface of claim 1, including a repository stored in memory accessible by at least one node in the network of document types for use in a plurality of transactions, and wherein the definition of one of the input and output documents includes a reference to a document type in the repository.
14. (Original) The method of claim 13, wherein the repository of document types includes a document type for identifying participant processes in the network.
15. (Original) The interface of claim 1, wherein the definitions of the input and output documents comprise document type definitions compliant with a standard Extensible

Markup Language XML.

16. (Original) The interface of claim 1, wherein the machine readable data structure including interpretation information comprises a document organized according to a document type definition compliant with a standard Extensible Markup Language XML.

17. – 60. (Cancelled).

61. (Original) A method for programming a commercial transaction in a network, comprising:

defining a machine readable definition of an input document for a node in the network including resources to execute a process in the transaction, and a machine readable definition of an output document for the node, the definitions of the input and output documents comprising respective descriptions of sets of storage units and logical structures for the sets of storage units; and

providing interpretation information for the logical structures to the node.

62. (Original) The method of claim 61, wherein the interpretation information includes data type specifications for at least one logical structure in the definitions of the input and output documents.

63. (Original) The method of claim 61, wherein the interpretation information includes at least one data structure mapping predefined sets of storage units for a particular logical structure in the definitions of the input and output documents, to respective entries in a list.

64. (Original) The method of claim 61, the step of providing interpretation information includes providing a repository in memory accessible by at least one node in the network storing a library of logical structures, and interpretation information for logic structures.

65. (Original) The method of claim 61, including defining a machine readable specification of an interface including a document compliant with a definition of an interface document including logical structures for storing an identifier of a particular transaction, and at least one of the definitions and references to the definitions of the input and output document.

66. (Original) The method of claim 61, wherein the storage units comprise parsed data.

67. (Original) The method of claim 66, wherein the parsed data in at least one of the input and output documents comprises:

character data encoding text characters in the one of the input and output documents, and

markup data identifying sets of storage units according to the logical structure of the one of the input and output documents.

68. (Original) The method of claim 67, wherein at least one of the sets of storage units encodes a plurality of text characters providing a natural language word.

69. (Original) The method of claim 67, wherein the interpretation information for at least one of the sets of storage units identified by a particular logical structure of at least one of the input and output documents, encodes respective definitions for sets of parsed characters.

70. (Original) The method of claim 66, wherein the storage units comprise unparsed data.

71. (Original) The method of claim 61, wherein the definitions of the input and output documents comprise document type definitions compliant with a standard Extensible Markup Language XML.

72. (Original) The method of claim 61, including:

providing a parser to generate event signals in response to logical structures in the definition of the input document; and

providing event listener programs which respond to the event signals to execute the process.

REMARKS

Claims 1-16 and 61-72 are pending in this action. They were previously rejected under 35 USC 103(a) as being unpatentable over McKendrick, *Banks begin to play with XML*, 11 Bak Technology News, Sep.1998, 6-7, in view of W3C, *Extensible Markup Language (XML) 1.0*, Feb. 10, 1998, 1-37. These rejections stand after appeal, subject to reconsideration in light of new evidence at the Board's invitation.

Appeal Recap

This case has been pending on appeal. The Board issued its Decision on Appeal on August 31, 2006 ("DoA").

During the appeal, only the independent claims 1 and 61 were separately argued. Accordingly, the Board did not address limitations of any of the dependent claims in its opinion.

Applicants responded to the opinion by filing an extensive Appeal Request for Rehearing ("RfR"), including copies of additional publications not previously of record. On May 21, 2007, the Board granted rehearing but denied the relief requested. The Board's Decision On Request for Rehearing ("DoRR") declined to consider the additional publications submitted for rehearing and invited Applicants to submit the new evidence to the Examiner with a request for continued examination.

We call the Examiner's attention to our filing of a petition and second request for rehearing. If the petition under Rule 183 to permit a second rehearing is granted, the Board may retain jurisdiction. The petition should have been decided by the time this RCE comes up on the Examiner's bi-weekly docket.

One of the positions asserted in the petition for second rehearing is that first rehearing was decided on new grounds and, therefore, the Decision On Request for Rehearing should not be considered final for purposes of judicial review.

Related Prosecution

This case is being prosecuted in parallel with application no. 09/633,365, pending before Examiner Kenneth Coulter. A number of rejections have been made and withdrawn. The latest rejection is dated February 8, 2007. The related case may be considered of interest to the Examiner, because of similar elements in the claims of the cases.

New Evidence that Strengthens the Declarations and Explains how the Declarations Would be Understood by One of Skill in the Art

The Board's original decision (DoA at 8-9) calls for more evidence about the "first draft of CBL" and treats the issue as one personal to these inventors. We responded by submitting Google searches and historical texts that explained how these inventors were educating those of ordinary skill in the art to understand "CBL" at the time this application was filed. We submitted:

- Glushko, *Implementing Domain-specific Commerce Languages with a Common Business Library* (delivered July 25, 1998) 32 pp.;
- Glushko et al., *An XML Framework for Agent Based E-Commerce*, Communications of the ACM, Vol. 42, No. 3, pp. 106-114 (Mar. 1999), 9 pp.;
- Glushko and McGrath, *Document Engineering: Analyzing and Designing Documents for Business Informatics and Web Services* (MIT Press 2005), 2 pp. excerpt;
- Sall, Kenneth B., *XML Family of Specifications: A Practical Guide* (Addison Wesley 2002), 1 pp.'

This evidence is resubmitted with this RCE and accompanied by:

- LaPlante, Glushko et al., *Document Standards & Technologies for Commerce Applications* (delivered Sept. 1, 1998) accessed at http://www.google.com/search?q=cache:hq5pefHRwksJ:seminars.seyboldreports.com/1998_san_francisco/ETAPE_26.html+%22cbl+1.0%22+veo&hl=en&gl=us&ct=clnk&cd=7 on Oct. 26, 2006, 28 pp.;
- xCBL.org, *About xCBL* (copyright 2000) accessed at <http://www.xcbl.org/about.shtml>, on July 20, 2007, 2 pp.;
- Allen, *Common Business Library (CBL)* (May 1999) accessed at <http://www.infoloom.com/gcaconfs/WEB/granada99/all.HTM> on July 20, 2007, 9 pp.;
- Bosak, *UBL Update*, OASIS Symposium on the Future of XML Vocabularies, Slide 3 (Apr. 25, 2005) viewed at www.oasis-open.org/events/symposium_2005/slides/bosak.pdf on July 20, 2007, 10 pp.;
- Meltzer and Glushko, *XML and Electronic Commerce: Enabling the Network Economy*, SIGMOND Record, Vol. 27, No. 4, 21-24 (Dec. 1998), 4 pp.;

This material approaches 100 pages of evidence to explain CBL and show that Glushko possessed an actual reduction to practice prior to McKendrick, as will be discussed below.

The Board (DoRR at 7) considered what we submitted to be new evidence and invited Applicants to submit it with an RCE. “We note that if Appellants wish to have the newly presented evidence considered by the Examiner, the proper procedure is to file a Request for Continued Examination (RCE) under 37 C.F.R. § 1.114.” Following this direction, the evidence is now of record and to be considered without any prejudgment from the Board.

Review of Glushko’s Slides, 1999 Article and 2005 Book

We used Google¹ to find a presentation by inventor Glushko that predates McKendrick, and explains use of CBL as asserted in the Veo status report, Exhibit A. We also located scholarly descriptions of these inventors’ work on CBL.

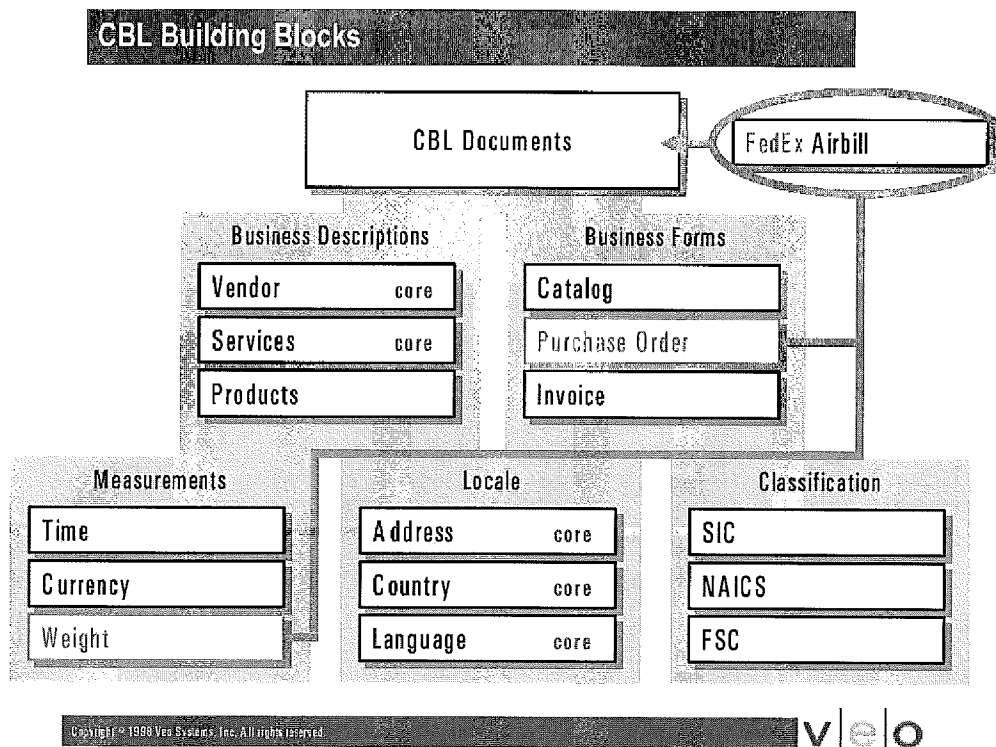
On July 25, 1998, the International Workshop on Component-based Electronic Commerce was held at the Fisher Center for Management & Information Technology, Haas School of Business, UC Berkeley. The conference program, accessed at <http://groups.haas.berkeley.edu/citm/conferences/cec/> on October 26, 2006, lists inventor Meltzer as the keynote speaker and a closing panelist. It lists inventor Glushko as a speaker.

Glushko’s slides use CBL document schemas in an interface definition data structure that includes input and output documents. Glushko, *Implementing Domain-specific Commerce Languages with a Common Business Library*, Slides 29-31 (delivered July 25, 1998) accessed at <http://groups.haas.berkeley.edu/citm/conferences/cec/Presentations/Session3/glushko.pdf> on October 26, 2006. We reproduce and explain three of the slides, attaching all of the slides as evidence.

Purchase orders and invoices are both mentioned in “CBL Building Blocks” slide 29, reproduced below, which illustrates how Federal Express might use CBL to create

¹ Commerce One declared bankruptcy years ago and liquidated its physical assets. The company’s design and programming records were not transferred to the present assignee of record, Open Invention Network (“OIN”). The mission of OIN to preserve the Linux eco-system against potential IP challenges is generally described at www.openinventionnetwork.com.

an XML version of its airbill by customizing a generic purchase order DTD with specific information about shipping weight.



This slide matches FIG. 10 of the application.

Slide 30, "Business Services Described Using CBL" (below) is an example of an interface definition data structure written in XML. The two service interfaces defined are named Submit Order and Track Order. The Submit Order service references an input "po.dtd" and an output "poack.dtd". These "dtd" files are document schemas. (Notice that this Submit Order service accepts a purchase order as input and returns an acknowledgement as output, not an invoice.)

Business Services Described Using CBL

```
<service>
<service.name>Order Service</service.name>
<service.location>www.veosystems.com/order</service.location>
<service.op>
  <service.op.name>Submit Order</service.op.name>
  <service.op.inputdoc>po.dtd</service.op.inputdoc>
  <service.op.outputdoc>poack.dtd</service.op.outputdoc>
</service.op>
< service.op>
  < service.op.name>Track Order</service.op.name>
  <service.op.inputdoc>request.track.dtd<service.op.inputdoc>
  <service.op.outputdoc>response.track.dtd<service.op.outputdoc>
</service.op>
</service>
```

This example matches LISTING 5 of the CD-ROM appendix to the revised (reformatted) specification, which appeared on page 45 of the original specification (before we moved source code to an appendix.) This code, placed in memory accessible to a node on a network, is an **actual reduction to practice** of some claims, as discussed below.

Glushko's actual reduction to practice, presented in public two months before McKendrick, was used to show those of ordinary skill in the art how to use CBL to define an interface. One exemplary interface specification data structure is for an order service that receives and acknowledges purchase orders.

Slide 31, "CBL status" (reproduced below) explains that CBL already had been used in demonstration applications. This corroborates the inventors' testimony in declaration paragraph 3 that documents and registries were working for their intended purpose.

CBL Status

- CBL v1.0 contains a few dozen DTDs and modules developed from analysis of ISO, ANSI X.12, other standards
- CBL currently being used by Veo Systems in demonstration applications (Project Seitai, GSA catalog interoperability)
- CBL to be starting “fodder” for CommerceNet-sponsored WG to develop open framework for interoperability of domain-specific commerce languages (just getting under way)

These three slides part of what Glushko used to present the new technology and explain how Veo was using CBL in interface specification data structures. With the benefit of Glushko's slides, one of ordinary skill in the art could read the status report and recognize that these inventors were using CBL as claimed, months before McKendrick's popular press report.

A 1999 article by inventors Glushko and Meltzer, written with Dr. Jay Tenenbaum of CommerceNet, also gives 1997 as the year for Veo's technology. Glushko, et al., *An XML Framework for Agent Based E-Commerce*, Communications of the ACM, Vol. 42, No. 3, pp. 106-114 (Mar. 1999). The authors explained their work, at 108: “Conceived originally as a CORBA-based interoperability framework, the eCo System architecture was **recast in 1997 on an XML foundation**, due to XML's simplicity and widespread adoption ...” This article explains Glushko slides 29 and 30, at pages 112-13.

In a textbook that he co-authored, Professor Glushko again referred to the 1997 work on CBL. “The earliest effort to attack the problem of semantic overlap among XML vocabularies for business applications was the XML **Common Business Library**, whose **first version was released in 1997**.” Glushko and McGrath, *Document Engineering: Analyzing and Designing Documents for Business Informatics and Web Services*, at 130 (MIT Press 2005)

Review of Sall's Book, the Seybold Transcript, xCBL.org, Allen and Bosak

A reference volume by Kenneth B. Sall, *XML Family of Specifications: A Practical Guide*, at 1073 (Addison Wesley 2002) provides the following glossary-like description of the xCBL e-commerce specification, which again mentions 1997:

Commerce One's XML Common Business Library promotes "cross-industry exchange of business documents such as product descriptions, purchase orders, invoices, and shipping schedules." Another goal is "to make the business documents, forms and messages that flow between businesses comprehensible to each business no matter what computer system is used." **The insightful CBL effort predates the XML 1.0 Recommendation, dating back to 1997.** xCBL uses a mature schema specification (SOX) which is a forerunner of the W3C's XML Schema Standard. See <http://www.xcbl.org/> for details.

(emphasis added). From this description, we see that "CBL" could be understood by those of skill in the art without any need to attach a copy of the first version of CBL to the declarations.

A transcript of a panel discussion on September 1, 1998 further mentions CBL and provides insights into the level of skill in the art. The Panel was part of the Seybold San Francisco/Publishing '98 Web Publishing Conference, entitled "Document Standards & Technologies for Commerce Applications". Glushko was careful about what he said and what he avoided talking about.

Mr. Glushko: ... So we've got to find ways to define those common document models for different communities of commerce. And that's what domain-specific languages are all about, so we're seeing a lot of activity here. ... This is a great idea but it's also a terrible idea. You see, XML makes easy-to-create markup languages. That's the good thing. The bad thing is that it makes it easy to create mark-up languages. And the value of a mark-up language or any language in general depends on the number of people that speak it ... **If we're not lucky, the whole world will invent its commerce language and we'll have the same problem of the Tower of Babel. We'll have no languages that are mutually intelligible.**

So **our vision** of how this works is as follows: Your company will publish on its Web site essentially a little corner that is the XML corner that will say here's how I want to do business electronically with you. I want to have a service, I'll call it the ordering service, that has two transactions. The first one is a submit transaction, the second one is a transaction, and basically, I'm saying **if you send me a document that conforms to po.dtd, a purchase order, I will send you back a purchase order acknowledgement.** And

these DTDs could be defined in some global registry or in some commerce registry or some other places all over the world--they should be freely available--that simply say how you want to do business automatically. Well, this is CBL. We have basically just released it to the world this week and after having used it in several demonstration projects, one with the federal government and one with the consortium of Japanese companies, and we have decided to find a way to make this most publicly available and most robust as possible.

Now we actually have a technology business wrapped around this, which I won't say very much about except that the idea is that if you have common business building blocks, you can define you own document models, expose them to the world, and then have a process which takes those things in and out of your legacy systems.

Audience: Is XML itself stable politically and otherwise?

Mr. Glushko: I sort of think XML is stable enough. There's a proposed recommendation from the W3C that came out in February that people are doing. I think there's always--you could always take a specification and try to aggressively implement or we could say let's use the sort of core stuff, which is not likely to change, and what we're trying to solve is agreement on basic tag sets to simplify the concept, and our library uses primitive concepts and actually doesn't push the envelope in terms of using XML as a specification. We're doing things that you could do on your Web site today with very common, easy tools. We're not pushing the envelope with XML. XML is undergoing a lot of changes. There's a major effort coming down the road which is an XML schema language, which would let you describe more data typing and semantic information inside of the XML definition, and that will be really important for commerce.

Another source of information about CBL is the organization xCBL.org, which has continued the development of standard business document interfaces even after Commerce One's demise. On their website, <http://www.xcbl.org/about.shtml>, viewed July 20, 2007, the following historical information appears:

The Evolution of xCBL

xCBL began its life at Veo Systems in 1997. At that time it was called simply CBL, and was a research project partly funded by the Department of Commerce's Advanced Technology Program. CBL was developed to test the limits of XML for e-commerce and to identify requirements for XML design, development, and transaction tools and platforms. Subsequently, Veo invented the first object-oriented XML schema language; SOX, the Schema for Object-Oriented XML; as a result of the lessons learned in the first version of CBL.

This confirms that CBL development was well under way prior to the May 11, 1998 date given in the declarations.

The inventor who was not available to sign declarations also published an account of the development of xCBL in 1999, accessed at <http://www.infoloom.com/gcaconfs/WEB/granada99/all.HTM>, on July 20, 2007:

I began work on CBL in August 1997. Version 1.1 was released in September 1998, and version 1.2, which is represented in both DTD (Document Type Definition) syntax and SOX, was completed in November 1998. At that point the specification had done its job in providing proof of concept, both to us and to the many who have downloaded the distribution; CBL is currently being employed only as a reference model. Further work is desirable to harmonize CBL semantics with those of EDI (both the X12 and EDIFACT flavors), and with other specifications that have appeared since my CBL work began. In the meantime, we've used a simplified subset of CBL for several successful demo projects.

The 1997 development date is recited yet again.

Jon Bosak, a legendary Sun Microsystems Distinguished Engineer who organized and led the working group that created XML, in April 2005 placed the release of CBL v1.0 as 1Q1998, consistent with the declarations. See, *UBL Update*, OASIS Symposium on the Future of XML Vocabularies, slide 3 (Apr. 25, 2005) viewed at www.oasis-open.org/events/symposium_2005/slides/bosak.pdf on July 20, 2007. (Interestingly, one of his slides shows the many steps and processes that are between order and invoice, in a system design.) Bosak's date for release of CBL v1.0 is repeated in various OASIS/UBL materials.

Application of the New Evidence to Understanding the Declarations

The evidence above establishes developmental work by Veo on CBL in 1997 (Glushko; Sall; xCBL.org; Allen), a release of v1.0 in early 1998 (Bosak), and general

release of v1.1 by early September 1998 (Allen; Glushko). The new evidence illustrates and explains CBL.

The evidence that illustrates CBL also establishes that Glushko showed an actual reduction to practice of a machine readable interface specification to transaction processes that provided a definition of an input document and a definition of an output document, in a single XML-coded data structure. He showed the public this actual reduction to practice months prior to the September publication of McKendrick, on July 25, 1998. We will read Slide 30 on the claims, below.

This evidence helps the Examiner understand the declarations and answers the Board's question, what was in the first draft of CBL? These inventors' actual reduction to practice is illustrated by Slide 30, which is found in this patent application as LISTING 5 of the CD-ROM appendix to the revised (reformatted) specification, which appeared on page 45 of the original specification (before we moved source code to an appendix.)

Reading Slide 30 on the Claims

One criticism that the Board had was that only the independent claims were addressed in our prior exercise of reading evidence (declarations) on the claims. This time, we'll walk through all of the claims.

Regarding use of evidence other than a Rule 131 affidavit to establish inventors' date of reduction to practice, we cite the statute § 102 and *Ex parte Foster*, 105 O.G. 261 (Comm'r Pat. 1903) (copy attached). We point out that MPEP § 715.04 cites *Ex parte Foster* as good authority, despite its early date. The statute only disqualifies claims from being granted if a reference predates the inventors' work, it does not limit the proof to Rule 131 or any particular evidence. The case *Ex parte Foster* makes it clear that Rule 131 is not an exclusive means of presenting evidence regarding inventors' work prior. The MPEP continues to endorse *Ex parte Foster*, even though it was decided 104 years ago. A publication by the inventors is a natural way to prove a date of invention prior to the application filing.

Claim 1: Slide 30 is a machine readable specification, even in a Powerpoint presentation. It was in a machine readable medium when the Powerpoint presentation was given. It defines an interface to transaction processes (1) for receiving a purchase order and responding with a purchase order acknowledgement and (2) requesting order

tracking and responding with status information. The PO and ACK are both XML documents, as are the request and response. The input document definitions are referenced by "po.dtd" and "request.track.dtd", which are data type definition (dtd) files. The output document definitions are referenced by "poack.dtd" and "response.track.dtd". The context of the slides and other remarks by Glushko including ongoing demonstration projects make it clear that this interface was hosted and accessible to a plurality of nodes on a network, in the development environment from which it was borrowed for the Powerpoint presentation. Therefore, Slide 30 removes McKendrick as a reference that can be applied against claim 1, because Slide 30 is an actual reduction to practice on public display before McKendrick.

Claim 2: The Examiner relies on the XML specification for limitations of claim 2 that go beyond claim 1. That specification was in use as part of Slide 30, so Slide 30 shows enough to remove McKendrick as a reference regarding claim 2.

Claim 3: The Examiner relies on the XML specification for claim 3 and that specification was in use as part of Slide 30, so Slide 30 shows enough to remove McKendrick as a reference regarding claim 7.

Claim 4: The Examiner relies on general principle for claim 4 and general principle is part of Slide 30, so Slide 30 shows enough to remove McKendrick as a reference regarding claim 4.

Claim 5: The identifier of a particular transaction appears in Slide 30 as "Submit Order" and "Track Order." The references to definitions of the input and output documents are the dtd file names.

Claim 6: The identifier of an interface coincides with the transaction name in Slide 30, appearing as "Submit Order" and "Track Order." The references to definitions of the input and output documents are the dtd file names.

Claim 7: The Examiner relies on either general principle or the XML specification for claim 7 and both general principle and that specification was in use as part of Slide 30, so Slide 30 shows enough to remove McKendrick as a reference regarding claim 3.

Claims 8-12: The Examiner relies on the XML specification for claims 8-12 and that specification was in use as part of Slide 30, so Slide 30 shows enough to remove McKendrick as a reference regarding claims 8-12.

Claim 13: The Examiner relies on general principle for claim 13, as she did for claims 4-5, and general principle is in use as part of Slide 30, so Slide 30 shows enough to remove McKendrick as a reference regarding claim 13.

Claim 14: The Examiner relies on the XML specification for claim 14 and that specification was in use as part of Slide 30, so Slide 30 shows enough to remove McKendrick as a reference regarding claim 14.

Claim 15: The dtd files referenced in Slide 30 are recognizable as compliant with a standard Extensible Markup Language XML.

Claim 16: The data structure that appears in Slide 30 is recognizable as compliant with a standard Extensible Markup Language XML.

Claim 61: Slide 30 is a machine readable specification, even in a Powerpoint presentation, which corresponds to the result of the defining step. It was in a machine readable medium when the Powerpoint presentation was given. It defines input and output documents. The PO and ACK are both XML documents, as are the request and response. The input document definitions are referenced by "po.dtd" and "request.track.dtd", which are data type definition (dtd) files. The output document definitions are referenced by "poack.dtd" and "response.track.dtd". The context of the slides and other remarks by Glushko make it clear that this definition was to be made available to a requesting node in a network. Therefore, Slide 30 removes McKendrick as a reference that can be applied against claim 61, because Slide 30 demonstrates reduction to practice of the claimed method and public display before McKendrick.

Claims 62-64: The Examiner relies on the XML specification for claims 62-64 and that specification was in use as part of Slide 30, so Slide 30 shows enough to remove McKendrick as a reference regarding claims 62-64.

Claim 65: The identifier of a particular transaction appears in Slide 30 as "Submit Order" and "Track Order." The references to definitions of the input and output documents are the dtd file names.

Claims 66-70: The Examiner relies on the XML specification for claims 66-70 and that specification was in use as part of Slide 30, so Slide 30 shows enough to remove McKendrick as a reference regarding claims 66-70.

Claim 71: The dtd files referenced in Slide 30 are recognizable as compliant with a standard Extensible Markup Language XML.

Claim 72: The Examiner relies on general principle for claim 72 and general principle is part of Slide 30, so Slide 30 shows enough to remove McKendrick as a reference regarding claim 72.

In this section, we have shown that Glushko Slide 30, which is new evidence of prior invention, can be read on each and every one of the pending claims. This evidence of prior invention removes McKendrick as a reference against these claims.

Reading the Declarations Combined with the New Evidence on the Claims

The Board invited Applicants to resubmit their evidence and arguments regarding how to read the declarations: “[I]f Appellants wish to have the newly presented evidence considered by the Examiner, the proper procedure is to file a Request for Continued Examination (RCE) under 37 C.F.R. § 1.114.” (DoRR at 7) This section proceeds as suggested and combines the Declarations with new evidence.

Because the Board did not understand the initials “CBL” (DoA at 8-9), which appear in Exhibit A, we have submitted nearly 100 pages of CBL-related evidence. It clarifies the following excerpt::

CNgroup has made substantial progress in both during the first quarter.

CBL (Common Business Language) enables semantic interoperation and integration of different commerce applications. CBL defines the metadata for making a business and its services a self-describing "eCo component"; it enables the intelligent query and aggregation of product catalogs and descriptions; it represents the forms and messages needed for commercial transactions; and it can be used to "wrap" formats and messages to make legacy applications "eCo-compliant". Specific technical activities performed during the first quarter as part of CBL R&D included:

1. Development of a "design philosophy" for overall scope and approach of CBL
2. Analysis of existing standards for common information types and semantic primitives. Where appropriate, semantics have been drawn from the UN/EDIFACT Basic Semantic Unit data dictionary and certain ISO and IETF standards (e.g., for geographical location, date and time, currency, weights and measures).
3. Analysis of proposed metadata frameworks for Internet resources (Dublin Core, RDF, MCF).
4. Analysis of semantics of commerce as embodied in EDI X12 transaction sets, Uniform Commercial Code, and in proposals like the Open Buying on the Internet specification.
5. Creation of first draft of CBL to support the requirements of Project Seitai (described above in "Project Baseline").
6. Determining an approach for CBL support of Industry applications (and for ATP demonstrations in particular).

The development of CBL has strongly shaped the requirements for the eCo runtime platform. XML is now at the core of the eCo architecture, and the eCo server can be thought of as an XML processing platform on which CBL is the reference application. The use of XML inside the eCo platform as well as in its applications has enabled the server to be more capable and extensible than we conceived at the time of the proposal.

In particular, the eCo server has now subsumed the registry and query services that had been envisioned as part of the Taxonomy of Everything in our proposal. The TOE was proposed as a scalable, distributed registry service for implementing Internet-based directory and translation services, and a key architectural building block and core task in the Phase 1 plan.

From the description in this excerpt of CBL, it is clear that the new evidence refers to the same CBL as the Exhibit A above, which predates McKendrick by several months. The Board specifically asked about the "first draft of CBL to support the requirements of Project Seitai," in numbered item 5. (DoA at 8-9; DoRR at 2) From the new evidence provided, we see that CBL was a way of defining a single document, for instance, a purchase order. CBL is one of the building blocks that led to the interface specification data structure illustrated in Slide 30. Project Seitai was well under way as a demonstration project by July 1998.

We focus the Examiner's attention on the last two paragraphs of the excerpt, which switch from CBL to the runtime platform. "XML is now at the core of the eCo architecture and the eCo server can be thought of as an XML processing platform on which CBL is the reference application. The use of XML inside the eCo platform as well as in its applications has enabled the server to be more capable than we conceived at the time of the proposal. [¶] In particular, the eCo server has now subsumed the registry and query services ..." What does this mean? First, work on CBL began in 1997 and the first draft of CBL was up and running in "1Q98". (Bosack) Second, CBL was a tool for defining a document such as a Purchase Order or PO Acknowledgement. (Glushko Slide 28) Third, XML was being used to define processing by the eCo server, which was recast on an XML foundation in 1997. (Glushko March 1999 at 108) Fourth, the eCo server included a registry. (Exhibit A) Fifth, Glushko was publicly showing a fragment of a registry entry in July 1998, an actual reduction to practice. (Glushko Slide 30)

It takes some decoding, which benefits from the wealth of new evidence, but this terse status report excerpt corroborates the declarations, which we will now read on the claims.

Claim 1: The declarations provide testimony, a form of evidence that a registry had been implemented and used in a method before March 11, 1998. The registry and method worked for establishing transactions among trading partners in a network. The specifications included definitions of input documents and output documents. The specifications were in machine readable memory and accessible upon request to nodes in a network. This reads the declaration evidence on claim 1. (The specifications also included services offered by the trading partners.)

The particular statements in the declaration are corroborated by Exhibit A in light of the new evidence. Implementation of the registry is specifically mentioned in Exhibit A, "the eCo server has now subsumed the registry". The registry is depicted in Glushko Slide 23. The registry is also described by Glushko in the September 1998 Seybold conference transcript, quoted above. CommerceNet, a non-profit with ties to Commerce One, began rallying members world wide around CBL version 1.1 in September 1998, per Meltzer 1998, p. 24. The declaration evidence that a registry was implemented is consistent with a great deal of corroborating evidence.

The registry used CBL, which is mentioned throughout Exhibit A. This use of the first draft of CBL is consistent with several sources, including the Glushko slides, Sall and Allen. Glushko Slide 31 recounts that the technology had been used in demonstration Project Seitai. Public reporting of a demonstration project using this technology in July corroborates the declaration testimony that an actual reduction to practice was accomplished prior to March 11, - a specific date for launch of Project Seitai is not necessary to lend credibility to the declaration, especially when the report was made at an industry conference. The declaration testimony that input and output document definitions were used in the registry is thoroughly corroborated, once CBL is understood.

Registry entries included both an input and output document, as an architecture for "loosely coupling" processes. The declarations make reference to both input and output documents, which is strongly corroborated by Glushko Slide 30 and LISTING 5 in the appendix of this application. Depictions and descriptions of a registry interface specification in Glushko Slide 23, In Glushko's September 1998 Seybold panel remarks and in Glushko 1999 at 113 consistently include both an input and output document in a registry entry. This goes beyond the single document descriptions which CBL provided. In Glushko 1999 at 108, use of XML in the registry is described as a significant improvement over attempts to develop an architecture using CORBA. This praise for using XML instead of CORBA is even stronger in Exhibit A, which recounts that "use of XML inside the eCo platform as well as in its applications has enable the server to be more capable and extensible than we conceived ..." Accordingly, the declaration testimony that registry entries included both an input document and output document is thoroughly corroborated.

Overall, there is much new evidence about the development timeline which corroborates the declaration testimony. The Examiner now has two ways to disqualify McKendrick as a reference against claim 1: Slide 30 is a public display of an actual reduction to practice and the declaration testimony now is thoroughly corroborated.

Claims 2-4: The Examiner relies on the XML specification for features of claims 2-4 that go beyond claim 1. The use of XML is highlighted by Exhibit A which is part of the declarations. All of the new evidence describes these inventors' early and ground breaking use of XML for e-commerce.

Accordingly, attachment of Exhibit A to the declarations shows enough to remove McKendrick as a reference regarding claims 2-4.

Claims 5-6: Definitions of business services offered are expressly called out in the declarations, in addition to input and output documents. As corroboration, the identifier of a business service appears in Slide 30 as "Submit Order" and "Track Order." Each general discussion of these inventors' work mentions identifying the transaction and service to be requested. In Exhibit A, support for query services implies an identification of a particular transaction or service being offered, given the new evidence.

Accordingly, the testimony in the declarations is corroborated by new evidence and Exhibit A sufficiently to remove McKendrick as a reference regarding claims 5-6.

Claim 7: The Examiner relies on either general principle or the XML specification for claim 7. Both general principle and the XML specification were in use as part of the declarations including Exhibit A, so the declarations and Exhibit A show enough to remove McKendrick as a reference regarding claim 7.

Claims 8-12: The Examiner relies on the XML specification for limitations of claims 8-12 that go beyond claim 1. The XML specification was in use as part of the declarations including Exhibit A, so the declarations and Exhibit A show enough to remove McKendrick as a reference regarding claims 8-12.

Claim 13: The Examiner relies on general principle for claim 13, as she did for claims 4-5, and general principle is in use as part of the declarations including Exhibit A, so the declarations and Exhibit A show enough to remove McKendrick as a reference regarding claim 13.

Claim 14: The Examiner relies on the XML specification for claim 14 and that specification was in use as part of the declarations including Exhibit A, so the declarations and Exhibit A show enough to remove McKendrick as a reference regarding claim 14.

Claim 15: The references to XML in Exhibit A are recognizable as compliant with a standard Extensible Markup Language XML.

Claim 16: The references to XML in Exhibit A are recognizable as compliant with a standard Extensible Markup Language XML.

Claim 61: This claim is easier than claim 1, because the declarations describe a method and this is a method claim. The declarations provide evidence that a registry had been implemented and used in a method. It worked for establishing transactions among trading partners in a network. Implementing the registry included defining machine readable specifications that included definitions of input and output documents. The specifications were in machine readable memory and accessible upon request to nodes in a network. This reads the declaration evidence on claim 61.

The particular statements in the declaration are corroborated by Exhibit A in light of the new evidence. Implementation of the registry is specifically mentioned in Exhibit A, "the eCo server has now subsumed the registry". The registry is depicted in Glushko Slide 23. The registry is also described by Glushko in the Sept. 1998 Seybold conference transcript, quoted above. CommerceNet, a non-profit with ties to Commerce One, is identified as an operator of an industry registry of service definitions, illustrated as including input and output documents in Gushko 1999 at 113. CommerceNet began rallying members world wide around CBL version 1.1 beginning in September 1998, per Meltzer 1998, p. 24. The declaration evidence that a registry was implemented is consistent with a great deal of corroborating evidence.

The registry used CBL, which is mentioned throughout the excerpt. This use of the first draft of CBL is consistent with several sources, including the Glushko slides and Allen. Glushko Slide 31 recounts that the technology had been used in demonstration Project Seitai. Public reporting of a demonstration project using this technology in July corroborates the declaration testimony that an actual reduction to practice was accomplished prior to March 11, - a specific date for launch of Project Seitai is not necessary to lend credibility to the declaration, especially when the report was made at an industry conference. The declaration testimony that definitions of documents were part of the registry is thoroughly corroborated, once CBL is understood.

Registry entries included both an input and output document, as an architecture for loosely coupling processes. The declarations make reference to both input and output documents, which is strongly corroborated by Glushko Slide 30 and LISTING 5 in the appendix of this application. Depictions and descriptions of a registry interface specification in Glushko Slide 23, Glushko's Sept. 1998 Seybold panel remarks and

Glushko 1999 at 113 consistently include both an input and output document in a registry entry. This goes beyond the single document descriptions on which CBL focused. In Glushko 1999 at 108, use of XML in the registry is described as a significant improvement over attempts to develop an architecture using CORBA. This praise for using XML instead of CORBA is even stronger in Exhibit A, which recounts that “use of XML inside the eCo platform as well as in its applications has enable the server to be more capable and extensible than we conceived ...” Accordingly, the declaration testimony that registry entries included both an input document and output document is thoroughly corroborated.

Overall, there is so much new evidence about the development timeline which corroborates the declaration testimony that the Examiner has two ways to disqualify McKendrick as a reference against claim 1: Slide 30 is a public display of an actual reduction to practice and the declaration testimony is now thoroughly corroborated.

Claims 62-64: The Examiner relies on the XML specification for features of claims 62-64 that go beyond claim 61. The use of XML is highlighted by Exhibit A which is part of the declarations. All of the new evidence describes these inventors’ early and ground breaking use of XML for e-commerce.

Accordingly, attachment of Exhibit A to the declarations shows enough to remove McKendrick as a reference regarding claims 62-64.

Claim 65: Definitions of business services offered are expressly called out in the declarations, in addition to input and output documents. As corroboration, the identifier of a business service appears in Slide 30 as “Submit Order” and “Track Order.” Each general discussion of these inventors’ work mentions identifying the transaction and service to be requested. In Exhibit A, support for query services implies an identification of a particular transaction or service being offered, given the new evidence.

Accordingly, the testimony in the declarations is corroborated by new evidence and Exhibit A sufficiently to remove McKendrick as a reference regarding claim 65.

Claims 66-70: The Examiner relies on the XML specification for claims 66-70. The XML specification was in use as part of the declarations including Exhibit A, so the declarations and Exhibit A show enough to remove McKendrick as a reference regarding claims 66-70.

Claim 71: The references to XML in Exhibit A are recognizable as compliant with a standard Extensible Markup Language XML.

Claim 72: The Examiner relies on general principle for features of claim 72 that go beyond claim 61. This general principle is part the declarations including Exhibit A, so the declarations including Exhibit A show enough to remove McKendrick as a reference regarding claim 72.

Considering McKendrick with New Evidence

At the Board's invitation, Applicants take this opportunity to present new evidence relevant the level of ordinary skill in the art and to the meaning and significance of the brief quotation attributed to Microsoft. The new evidence includes:

- *Registries and Repositories – XML/SGML Name Registration* (about Nov. 1998) accessed at <http://xml.coverpages.org/registryColl.html> on October 26, 2006, 30 pp.
- Microsoft Corp., *XML: Enabling Next-Generation Web Applications* (Apr. 3, 1998) accessed at <http://msdn.microsoft.com/archive/default.asp?url=/archive/en-us/dnarxml/html/xmlwp2.asp> on July 21, 2007, 15 pp.
- Bosworth, General Manager, Microsoft Corp., *Europe '98, Microsoft's Vision for XML* (May 18-21, 1998) viewed at <http://xml.coverpages.org/bosworthXML98.html> on July 21, 2007, 10 pp.
- Winer, *XML-RPC for Newbies* (July 14, 1998) viewed at <http://www.scripting.com/davenet/1998/07/14/xmlRpcForNebies.html> on July 21, 2007, 6 pp.
- Walsh, *Microsoft spearheads protocol push*, InfoWorld Electric (July 10, 1998) viewed at <http://infoworld.com/cgi-bin/displayStory.pl?98071.whsoap.htm> on July 21, 2007, 2pp.
- Merrick et al., US 7,028,312, *XML Remote Procedure Call (XML-RPC)*.

We also will rely on some of the evidence previously discussed, as it bears on the level of ordinary skill.

Ex Parte Jud Explains how to Weigh Level of Ordinary Skill Evidence

First, we direct the Examiner's attention to *Ex parte Jud*, Appeal No. 2006-1061 (Jan. 30, 2007) (expanded panel, informational opinion) (copy attached), which discusses the level of ordinary skill in the art. Designation of this opinion as "informational" makes it persuasive, if not binding. The informational designation was created to streamline the procedure for designating noteworthy opinions. It is easier for the Chief Administrative Judge to designate an opinion as informational than as precedential, which requires an affirmative vote by a majority of the active judges of the BPAI, which recently was 59 judges.

In *Ex parte Jud*, at 2-3, the Board reiterated the *Graham* factors for nonobviousness analysis, which include the level of ordinary skill in the art.:

The Supreme Court has elaborated that:

Under § 103, the scope and content of the prior art are to be determined; differences between the prior art and the claims at issue are to be ascertained; and the **level of ordinary skill in the pertinent art** resolved. Against this background, the obviousness or nonobviousness of the subject matter is determined. Such secondary considerations as commercial success, long felt but unsolved needs, failure of others, etc., might be utilized to give light to the circumstances surrounding the origin of the subject matter sought to be patented. *Graham v. John Deere Co.*, 383 U.S. 1, 17-18 (1966). These four determinations have come to be known as the Graham factors.

The Board discussed the value that it finds in three categories of evidence: the application itself, publications and testimony.

The Board afforded the applicant's disclosure significance for its level of detail. Regarding the application, at 4:

The disclosure is particularly helpful when it describes the background to the invention and the applicant's contribution to the art. Care must be exercised, however, to ensure that the applicant's contribution is not itself mistaken as an admission regarding the pre-existing knowledge and skill in the art.

The Board goes on to comment that what the applicant disclosed in order to enable the new technology helps to determine the level of ordinary skill in the art. *Id.* at 4-5. An enabling disclosure gives a sense of how much teaching it takes for one of ordinary skill in the art to understand and practice the new technology, establishing at least a floor on the level of skill in the art. Accordingly, a short disclosure suggests an easily understood

innovation; a long disclosure suggests a new development that must be carefully explained in order for one of ordinary skill to understand.

The second category is documentary evidence or references, at 5.

References are typically indirect in their teachings regarding the skill level in the art. Moreover, the teachings may sometimes be incomplete since explaining the skill level in the art is rarely the intended purpose of a reference. References are generally entitled to great weight, however, because they are almost always prepared without regard to their use as evidence in the particular examination in which they are used.

Judges and trial lawyers similarly emphasize the value of evidentiary documents prepared for reasons unrelated to a dispute. Here, the Glushko references relied upon above and the Microsoft-related documents discussed below were prepared for reasons unrelated to this patent application.

The least helpful category of evidence is testimony about the education level and work experience of an artisan. *Id.* at 5-6.

Review of Applicants' Disclosure, Conference and Technical Committee Participation

These Applicants' devoted 113 pages to their disclosure plus 16 sheets of figures. Considerable detail is provided to teach those of ordinary skill how to practice new innovation. From public appearances, these inventors had a good basis for understanding how their disclosure would be understood by those of ordinary skill.

These inventors were of extraordinary skill and in demand as speakers at conferences and leaders of technical committees. At the July 25, 1998 conference, *International Workshop on Component-based Electronic Commerce*, held at Fisher Center for Management and IT Haas School of Business, University of California Berkeley, inventor Meltzer was the Key-Note speaker and one of the Closing Panel members who addressed, "A Marketplace for EC Components - What is missing?" See, <http://groups.haas.berkeley.edu/citm/conferences/cec/> viewed Oct. 26, 2006. Inventor Glushko spoke immediately following the conference lunch. Similarly, inventor Glushko was a panelist at the Seybold conference on September 1, 1998, from which we have submitted a transcript. Inventor Glushko is now an Adjunct Professor in the School of Information and Director of the Center for Document Engineering at UC Berkeley. See <http://www.ischool.berkeley.edu/~glushko/>. Inventor Terry Allen also spoke and wrote articles. He served with Jon Bosak of Sun Microsystems on an organizing committee of

the industry Organization for the Advancement of Structured Information Standards ("OASIS") (www.oasis-open.org) and was the first chairperson, in 1999, of the Registry & Repository Technical Committee. That committee developed a draft registry specification during 1999, which it released on December 8, 1999. See, http://www.oasis-open.org/html/oasis_news_archive/oasis_draft_spec.php. It would be a mistake, contrary to *Ex parte Jud*, to treat writings by these inventors as indicating the typical level of creativity of those of ordinary skill in the art in 1998, when they were acknowledged innovators and leaders in this field of development.

The level of **ordinary** skill in the nascent art of XML web services corresponds to the audience at these conferences, not the speakers (our inventors.) Recall the audience skepticism about XML on September 1, 1998:

Audience: Is XML itself stable politically and otherwise?

Mr. Glushko: I sort of think XML is stable enough. There's a proposed recommendation from the W3C that came out in February ...

What the panelists were teaching was what they thought those of ordinary skill did not already know. What they declined to talk about (e.g., Glushko not talking about the business they were developing around the technology) implies what those of ordinary skill did not know and did not learn at the conference.

The overall influence of these inventors on an emerging software architecture is reflected in the *Registries and Repositories* document which purports to give a reasonable overview of what people were thinking about in terms of XML registries in about November 1998. See, Note/Caution at 1. Among the first nine efforts listed on page 1 of the document related to registries, Veo and these inventors were involved in six of the nine. Part of Veo's early efforts were funded by (1) NIST. Veo was a subcontractor and participant in the NIST grant under the sponsorship of CommerceNet, which also sponsored efforts (3)-(5) in the list. Veo Systems appears as (6) in the list. Inventor Terry Allen chaired the repository committee for (9) OASIS. Therefore, one should not underestimate how involved and influential these inventors were evangelizing for their new web services document interface technology.

The combination of how much detail was considered necessary in October 1998 to enable this technology, the extraordinary level of skill in the art of these inventors and

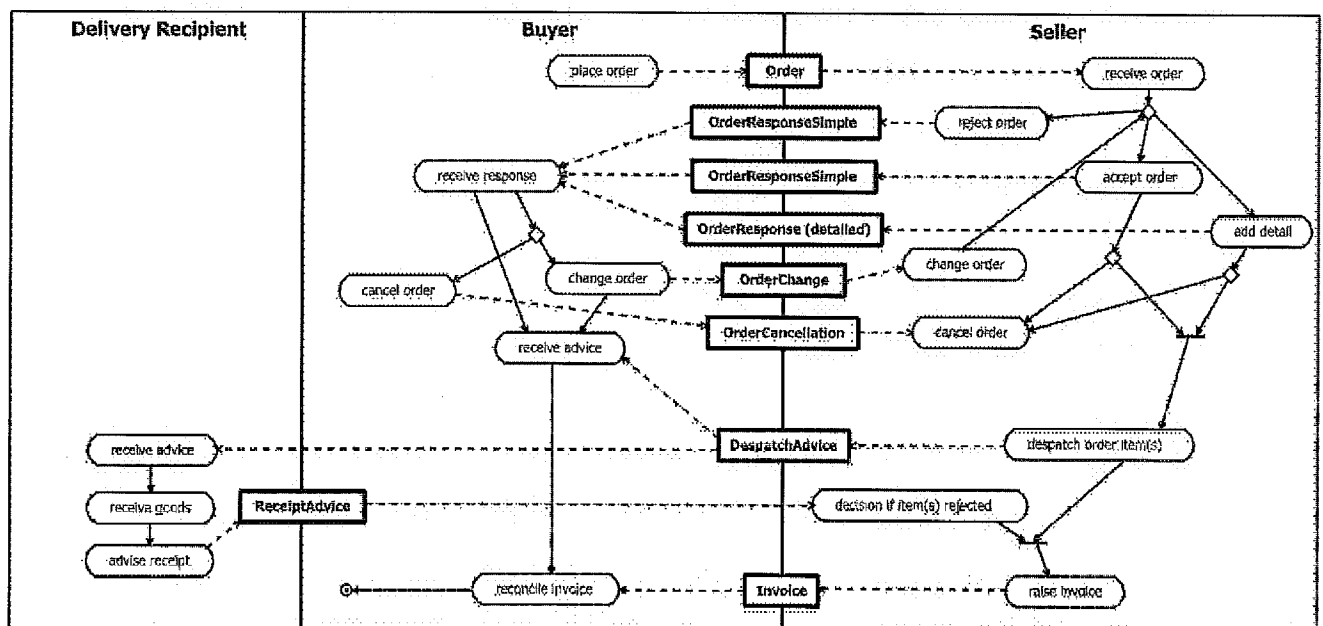
what they explained in the application but not the conferences informs our view of the level of ordinary skill in the art, following *Ex parte Jud*.

The References

In this section we use the new evidence to explore three issues: how purchase orders and invoices are understood in the industry to be connected; what those of ordinary skill in the art would have understood from Microsoft's 1998 comments on XML for ecommerce, some of which are excerpted in McKendrick; and how limited the McKendrick comment is, in light of the alternative architecture that Microsoft was advocating.

We see how the processing connection that links purchase orders and invoices came to be understood by the industry in Bosak Slide 5 (2005):

UBL order-to-invoice



Bosak's presentation given seven years after this application was filed depicts an industry consensus in an UBL order-to-invoice workflow template regarding a set of transactions, inputs and outputs that typically span the process space from a buyer placing an order to a seller sending an invoice. This evidence belies the assumption that a purchase order would be an input and an invoice would be an output to a

transaction defined by a machine readable specification that includes definitions of input and output documents. Common experience with HTML leads us to believe that a product selection screen and a checkout screen are analogous to a purchase order and an invoice. According to this new evidence, business purchasing transactions are much more complicated than consumer Internet shopping. When an XML-style interface is justified for purchasing activities, the use case is typically as Bosak depicts.

Next, we followed through on McKendrick's reporting and found a Microsoft archive document that includes the quoted phrase: The publication *XML: Enabling Next-Generation Web Applications* (April 3, 1998), at 9, includes the passage quoted by McKendrick, "Customer services are now migrating to Web sites from call centers and physical locations and will therefore benefit from the robust functionality of XML. And, because most of these business applications involve manipulation or transfer of data and database records, such as purchase orders, invoices, customer information, appointments, maps and such, XML will revolutionize end-user possibilities on the Internet by allowing a rich array of business applications to be implemented." This 15-page document is submitted for the Examiner to review. It does not teach our claimed architecture and does not actually teach software architecture. One of XML tool companies, listed at 14, took the next teaching step and at least labeled Microsoft's approach.

Dave Weiner of UserLand Software posted an article on July 14, 1998 entitled *XML-RPC for Newbies*. In this context, RPC means "remote procedure call". Winer reports that Microsoft was promoting RPC as "the XML-based protocol [that] fits into its goal..." *Id.* at 1. Microsoft was proposing XML-RPC as a way of marshalling parameter lists for procedure calls (at 2-3); it was not using XML business documents. In this article, Weiner relied in turn on a report by Jeff Walsh, *Microsoft spearheads protocol push* (July 10, 1998), which tied RPC to the first generation of SOAP. "The Simple Object Access Protocol, or SOAP, enables Remote Procedure Calls (RPCs) to be sent as Extensible Markup Language (XML) syntax across the Web's HTTP architecture. [¶] The protocol was developed by Microsoft, UserLand Software and DevelopMentor ..."

Microsoft's teaching of XML-RPC architecture during 1998, which was vaguely reported by McKendrick, is confirmed in remarks given by Microsoft General Manager Adam Bosworth as the Opening Keynote Address of SGML/XML '98, May 18-21, 1998

in Paris. His remarks were entitled *Europe '98, Microsoft's Vision for XML*, which makes them particularly relevant to understanding McKendrick. At 4, RPC is first mentioned.

"As soon as one tries to use XML even for something really simple like transporting the arguments that an RPC implies ..." Microsoft's teaching builds from there. On 5:

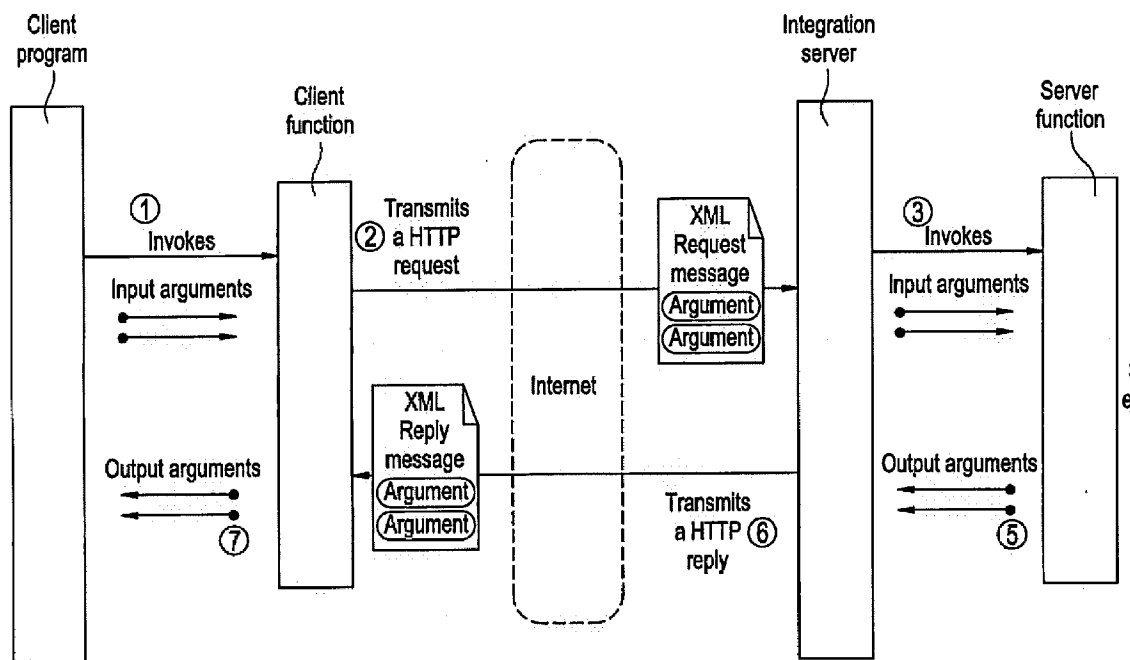
Let's imagine that the site isn't willing to support a general query language no matter how limited. What it is willing to do is expose certain URLs which when sent the appropriate parameters of Title or Author return a list of books for those Titles or Authors or both. Notice that **this is basically RPC**. Now, even in this case, the search engine would want to send the parameters to the book-purveying site in a simple, easy to engineer manner. If the "Site description" grammar assumed above **describes the desired shape of the parameters of the "method" and there is a standard grammar for marshalling parameters for RPC** or the "Site description" grammar describes the grammar of the search request, then the search engine knows what XML to send. Ideally, this **grammar for RPC** will also describe how synchronously the results may be returned and by what mechanism to allow for delayed return.

To summarize, Microsoft's GM indicated the need for several new XML grammars, with RPC possibly as an extension of Microsoft's pending XML-DATA proposal, at 6. Fifth in the list of new grammars to be developed and necessary to adoption of the emerging technology was, "Either an XML RPC proposal or extensions to the XML-IDL proposal above to describe the grammar of parameters submitted in an RPC and requisite envelope information".

Under "Predictions", at 9, Mr. Bosworth pointed developers toward a Microsoft approach that diverges sharply from the approach taken in our disclosure. "XML will become a widespread solution to interoperable RPC on the net." In boldface, the General Manager predicted, "**The programming model for building applications will change to:** One in which XML is the standard message format used for transmitting data and for RPC." *Id.* at 10 (bold facing in script for remarks).

From the new evidence, it is clear that one who explored McKendrick's vague quotation of a Microsoft document would find Microsoft advocating change in the programming model to use XML as a format for marshalling arguments and making remote procedure calls (RPC argument list.) With a bit of historical perspective, those of skill in the art (post-1998) should all agree that XML-RPC argument lists are not the same as document-style interface specifications for web services.

Regarding the nature of XML-RPC argument lists, we offer as new evidence a patent, Merrick et al. US 7,028,312 ("Merrick"), assigned to webMethods, entitled "XML Remote Procedure Call (XML-RPC)" and the provisional applications that preceded it, 60/079,100 filed March 24, 1998 and 60/096,909, filed August 17, 1998. We need not spend any time on the '100 provisional, because it does not include enough detail to afford an early priority date to any interesting disclosures in the '312 patent. In the '312 patent, we see an enabling disclosure of XML-based RPC technology, which issued as a patent. The level of detail provided in order to enable XML-RPC is much closer to the detail in our application than it is to McKendrick's 33 words. FIGS. 6 and 7, respectively, reflect the CORBA and XML-RPC approaches to remote procedure calls. The XML-RPC approach was patentable over CORBA, so we reproduce FIG. 7:



This figure depicts use of XML for argument passing. To see a format used for argument passing, which does not resemble the Business Information Documents (BIDs) disclosed in this application, we look to column 19:

```

<RPC TYPE="REQUEST">
  <VALUE NAME="accountID" TYPE="int">2001</
    VALUE>
  <VALUE NAME="zodiacSign">Aquarius</VALUE>
</RPC>
<RPC TYPE="REPLY">
  <VALUE NAME="orderNumber"
    TYPE="int">38553</VALUE>
  <VALUE NAME="fortune">XML is good for RPC</
    VALUE>
  <VALUE NAME="balance" TYPE="float">65.00</
    VALUE>
</RPC>

```

This passage illustrates a parameter list, rather than an input document and an output document. Over time, this use of XML has come to be called "rpc style", which is contrasted to "document style". Remote procedure calls have come to be considered more "tightly coupled" and document style interfaces more "loosely coupled." The code example, above, but not FIG. 7, is found in the earlier '909 provisional, so the code example but not the figure predates our filing. However, even the code example was submitted to the PTO after Glushko's conference and presentation and slides on July 25, 1998.

From the new evidence, one gets a better understanding of where Microsoft was leading and what one of ordinary skill in the art could discern from Microsoft's press. Of course, a programmer would not stop with McKendrick, because McKendrick teaches nothing about software architecture. Nor would a programmer be satisfied with the report from which McKendrick quotes, because it also is fluff. A person of ordinary skill would have looked for descriptions by Microsoft and its partners of any proposed new XML architecture that was under development.

The evidence is clear that Microsoft was leading those in the art towards use of XML for remote procedure calls, not towards a document-oriented interface. The postings and press make this clear, including a posting by an identified Microsoft partner. Significantly, RPC was the only approach that Microsoft General Manager Bosworth presented at a worldwide conference in May 1988. In the hierarchy of corporate titles, the Examiner should recognize that a General Manager ranks above a Director and that, in smaller organizations, one person often holds the titles President

and General Manager. While GM Bosworth was not Bill Gates, he was so highly placed in Microsoft's organization as to speak with authority on behalf of the corporation.

Let's apply this evidence to McKendrick. The first XML specification was released in February 1998, six months before McKendrick. McKendrick suggests a direction for new development, rather than providing an enabling disclosure – the contemporaneous patent applications provided extensive detail in order to enable the RPC style and document style interfaces. Persons of ordinary skill, at a September 1, 1998 conference asked the expert panel whether XML was stable enough, politically and otherwise, for them to use. Microsoft was suggesting use of RPC, but the details of XML for remote procedure calls were exposed by a Microsoft partner in a patent application filed in March 1999, that resulted in an issued patent. While the general notion of using XML for RPC was voiced as something to try, it was novel enough in fall 1998-spring 1999 to be patentable and a patent actually was granted. When one looks at XML-RPC, one sees a remote procedure call protocol that is much different than the claimed document interface specification, an alternative software architecture.

With the benefit of the new evidence and taking into account how one of ordinary skill in the art would respond to a 33-word report of Microsoft's new direction, it is clear that Microsoft was leading those of ordinary skill in a different direction. One of ordinary skill in the art would not find McKendrick to be an enabling disclosure of the claimed interface specification or even a suggestion in the direction that these inventors led, away from Microsoft's approach. The development efforts under way in 1998 were ground breaking and inventive, including work both on XML-RPC and these inventors' work on a document interface. These inventors were of extraordinary skill, they were the people to whom the industry looked to set a new direction. They set a new direction that has been widely accepted. The direction they set, which is reflected in these claims, is patentable over a 33-word report of Microsoft's other direction.

CONCLUSION

To review, we have presented considerable new evidence at the invitation of the Board, to be reviewed without prejudice because the Board did not consider any of it.

McKendrick is removed as a reference, both by the newly presented July 25, 1998 actual reduction to practice in Slide 30 and by the combination of new evidence with the declarations.

Moreover, the new evidence regarding the levels of ordinary skill in the art are where Microsoft actually was steering development makes it clear that McKendrick's 33 words do not enable the new technology claimed in a 129-page application.

Applicants respectfully submit that the pending claims are now in condition for allowance and thereby solicit acceptance of the claims as now stated.

Applicants would welcome an interview, if the Examiner is so inclined. The undersigned can ordinarily be reached at his office at (650) 712-0340 from 8:30 a.m. to 5:30 p.m. PST, Monday through Friday, and can be reached at his cell phone at (415) 902-6112 most other times.

Fee Authorization. The Commissioner is hereby authorized to charge underpayment of any additional fees or credit any overpayment associated with this communication to Deposit Account No. 50-0869 (OIN 1004-1). A duplicate copy of this authorization is enclosed.

Respectfully submitted,

Dated: July 23, 2007

/Ernest J. Beffel, Jr./
Ernest J. Beffel, Jr.
Registration No. 43,489

Haynes Beffel & Wolfeld LLP
P.O. Box 366
Half Moon Bay, CA 94019
Telephone: (650) 712-0340
Facsimile: (650) 712-0263

EVIDENCE APPENDIX

Each of the documents listed has been separately uploaded into Private PAIR as Non-Patent Literature.

Allen, *Common Business Library (CBL)* (May 1999) accessed at <http://www.infoloom.com/gcaconfs/WEB/granada99/all.HTM> on July 20, 2007, 9 pp.

Bosak, *UBL Update*, OASIS Symposium on the Future of XML Vocabularies, slide 3 (Apr. 25, 2005) viewed at www.oasis-open.org/events/symposium_2005/slides/bosak.pdf on July 20, 2007, 10 pp.

Bosworth, General Manager, Microsoft Corp., *Europe '98, Microsoft's Vision for XML* (May 18-21, 1998) viewed at <http://xml.coverpages.org/bosworthXML98.html> on July 21, 2007, 10 pp.

Glushko, *Implementing Domain-specific Commerce Languages with a Common Business Library* (delivered July 25, 1998), 32 pp.

Glushko et al., *An XML Framework for Agent Based E-Commerce*, Communications of the ACM, Vol. 42, No. 3, pp. 106-114 (Mar. 1999), 9 pp.

Glushko and McGrath, *Document Engineering: Analyzing and Designing Documents for Business Informatics and Web Services* (MIT Press 2005), 2 pp. excerpt.

LaPlante, Glushko et al., *Document Standards & Technologies for Commerce Applications* (delivered Sept. 1, 1998) accessed at http://www.google.com/search?q=cache:hq5pefHRwksJ:seminars.seyboldreports.com/1998_san_francisco/ETAPE_26.html+%22cbl+1.0%22+veo&hl=en&gl=us&ct=clnk&cd=7 on Oct. 26, 2006, 28 pp.

Merrick et al., US 7,028,312, *XML Remote Procedure Call (XML-RPC)*.

Meltzer and Glushko, *XML and Electronic Commerce: Enabling the Network Economy*, SIGMOND Record, Vol. 27, No. 4, 21-24 (Dec. 1998), 4 pp.

Microsoft Corp., *XML: Enabling Next-Generation Web Applications* (Apr. 3, 1998) accessed at <http://msdn.microsoft.com/archive/default.asp?url=/archive/en-us/dnarxml/html/xmlwp2.asp> on July 21, 2007, 15 pp.

Registries and Repositories – XML/SGML Name Registration (about Nov. 1998) accessed at <http://xml.coverpages.org/registryColl.html> on October 26, 2006, 30 pp.

Sall, Kenneth B., *XML Family of Specifications: A Practical Guide* (Addison

Wesley 2002), 1 pp.

Walsh, *Microsoft spearheads protocol push*, InfoWorld Electric (July 10, 1998) viewed at <http://infoworld.com/cgi-bin/displayStory.pl?98071.whsoap.htm> on July 21, 2007, 2pp.

Winer, *XML-RPC for Newbies* (July 14, 1998) viewed at <http://www.scripting.com/davenet/1998/07/14/xmlRpcForNebies.html> on July 21, 2007, 6 pp.

xCBL.org, About xCBL accessed at <http://www.xcbl.org/about.shtml>, on July 20, 2007, 2 pp.



UNITED STATES PATENT AND TRADEMARK OFFICE

OIN 1004-1
EJ8

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/173,858	10/16/1998	BART ALAN MELTZER	OIN 1004-1	4734

22470 7590 01/23/2008
HAYNES BEFFEL & WOLFELD LLP
P O BOX 366
HALF MOON BAY, CA 94019

EXAMINER

HUYNH, CONG LAC T

ART UNIT PAPER NUMBER

2178

MAIL DATE DELIVERY MODE

01/23/2008

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

HAYNES BEFFEL & WOLFELD LLP

JAN 25 2008

RECEIVED

HAYNES BEFFEL & WOLFELD LLP

DOCKETED: 1-25-08 BY: *SR*
ACTION: *US - Final Office Action & NOA*
DUE DATES: *3-23-08 - 2 MO Due Date*
4-23-08 - Due
7-23-08 - Final
DOCKET NO: *OIN 1004-1*

R-270

Office Action Summary	Application No.	Applicant(s)	
	09/173,858	MELTZER ET AL.	
	Examiner	Art Unit	
	Cong-Lac Huynh	2178	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 23 July 2007.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-16 and 61-72 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-16 and 61-72 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date <u>7/23/07</u> . | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. Applicant is hereby noted that the previous office action mailed on 10/3/07 is vacated. This action is responsive to the RCE filed on 7/23/07 to the application filed on 10/16/98.
2. Claims 1-16, 61-72 are pending in the case. Claims 1 and 61 are independent claims.
3. Claims 1-16 and 61-72 are under the principle of Res Judicata. The examiner's rejection of the above claims was affirmed at the Board of Appeals on 8/31/06. The instant claims have not been amended and are a duplicate of the claims presented earlier to the Board of Appeals. See MPEP 706.03 (w).

The issues decided by the Board are the same as those presented here, namely whether appellants had possession of the invention before the critical date by presenting evidence purportedly demonstrating an actual reduction to practice prior to March 11, 1998. The Board decided this issue stating, "After careful consideration of the evidence before us, we conclude that appellants have *failed* to provide a factual showing that the embodiment relied upon actually worked for its intended purpose as required to demonstrate an actual reduction to practice.

Furthermore, we find that appellants have failed to completely read the language of the instant claims on the proffered Exhibit A evidence provided in the declaration. In particular, we note that appellants have failed to point to specific portions of Exhibit A that demonstrate actual reduction to practice of the instant claimed "interpretation information providing a definition of an input document, and a definition of an output

document, the definitions of the input and output documents comprising respective descriptions of sets of storage units and logical structures for the sets of storage units" [claim 1, emphasis added]. While we recognize that XML (i.e., Extensible Markup Language) may provide Document Type Definitions (see e.g., W3C, p. 9, §2.8), the mere use of XML does not disclose input and output documents, per se. We have considered appellants' argument that the "forms and messages" disclosed in Exhibit A correspond to "input and output messages or documents" as part of CBL [brief, page 8, emphasis added]. However, we note that appellants have *failed* to provide a copy of the CBL first draft to be considered as evidence of actual reduction to practice.

We recognize that an accompanying exhibit need not support all claimed limitations, provided that any missing limitation is supported by the declaration itself. *Ex parte Ovshinsky*, 10 USPQ2d 1075 (Bd. Pat. App. & Inter. 1989) [emphasis added]. In the instant case, we note that appellants have also failed to explain how dependent claims 2-16 and 62-72 are supported by Exhibit A. In the alternative, appellants have *failed* to provide support for the missing dependent claim limitations in the declaration itself, so as to conclusively show possession of the complete instant claimed invention before the critical date. For at least the aforementioned reasons, we conclude that the character and weight of the evidence submitted pursuant to 37 C.F.R. § 1.131(b) is insufficient to show actual reduction to practice before the critical date of Sept. 1998. Accordingly, we consider the McKendrick reference as prior art, *infra*."

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identical disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

6. Claims 1-16, 61-72 remain rejected under 35 U.S.C. 103(a) as being unpatentable over McKendrick, *Banks begin to play with XML*, Bank Technology News, Sep 1998, Vol. 11, Iss. 9, pg. 6, 2 pgs, in view of W3C, *Extensible Markup Language (XML) 1.0*, 2/10/98, pages 1-37 (from the IDSs).

Regarding independent claim 1, McKendrick discloses:

- a machine-readable specification of an interface to transaction processes stored in memory accessible by at least one node in the network, including interpretation information providing a definition of an input document, and a definition of an output document (pages 1-2: McKendrick discloses applying XML in financial area to provide better bank services and utilizing XML for on-line business transactions involved with manipulation and transfer of data in the Internet such as purchase orders, invoices, and customer information. The purchase orders are considered as input documents, and the invoices are considered as output documents of the purchase orders in business transactions. Since the purchase orders as well as the invoices, which are the input and output documents, are in XML, they definitely include information providing the definition for such a document according to XML structures. And since the transaction documents are in XML format, these documents are machine-readable documents and should be stored in memory of a server accessible by at least one node in the network)

McKendrick does not explicitly disclose that the definitions of the input document and the output document comprising respective descriptions of sets of storage units and logical structures for the sets of storage units.

W3C discloses that each XML document comprises respective descriptions of set of storage units and logical structures for the set of storage units (page 3, Introduction: "XML documents are made up of *storage units* called entities, which contain either *parsed or unparsed data*. Parsed data is made up of characters, some of which form

character data, and some of which form *markup*. *Markup encodes a description of the document's storage layout and logical structure*. XML provides a mechanism to impose constraints on the storage layout and logical structure.”)

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to have combined McKendrick into W3C for the following reason.

McKendrick discloses the transaction documents such as the purchase orders and the invoices in XML format for a business transaction over the Internet where a user can search and buy an item on-line, and W3C discloses the structures of an XML document which comprises storage units and the logical structures for the set of storage units. This motivates to combine W3C into McKendrick for supporting the business transaction documents in XML format using the XML characteristics disclosed in W3C.

Regarding claim 2, which is dependent on claim 1, McKendrick does not disclose that the interpretation information includes data type specification for at least one logical structure in the definitions of the input and output document.

W3C discloses that each XML document contains one or more elements which are delimited by starts-tags and end-tags, and each element has a *type* identified by name called generic identifier and may have a set of *attribute specification* (page 13, Logical structure).

As mentioned in claim 1, since the documents used in the purchase transaction in McKendrick are in XML format, these documents inherit the features of a general XML

document as disclosed in W3C. This is applied for all the claims relating to the transaction document structures and W3C is used for rejecting.

Regarding claim 3, which is dependent on claim 1, W3C discloses that the interpretation information includes at least one data structure mapping predefined sets of storage units for a particular logical structure in the definition of the input and output documents, to respective entries in a list (pages 14-17).

Regarding claims 4 and 5, which are dependent on claim 1, McKindrick and W3C do not disclose explicitly that a repository in memory accessible by at least one node in the network storing a library of logical structures, interpretation information for logical structures, and the identifier of a transaction. However, it would have been obvious to one of ordinary skill in the art at the time of the invention was made to have modified McKindrick and W3C to include a repository in memory for storing logical structures and the identifier of a transaction interface since it was well known in the art that any defined data for a program in a network should have a name for identifying and should be stored in a memory of a server for using later on such as retrieving data, identifying data, or manipulating data.

Regarding claim 6, which is dependent on claim 1, W3C discloses that the machine readable specification includes a document compliant with a definition of an interface document including logical structures for storing an identifier of the interface, and for

storing at least one of specifications and references to specifications of a set of one or more transactions supported by the interface (page 13).

Regarding claim 7, which is dependent on claim 6, McKindrick does not disclose a reference to a specification of a particular transaction, and the specification of the particular transaction includes a document including logical structures for storing at least one of definitions and references to definitions input and output documents for the particular transaction. Instead, McKindrick discloses *applying XML for business-to-business transaction where data such as purchase orders and invoices are manipulated and transferred over the Internet* (page 2).

W3C discloses that each XML document comprises respective descriptions of set of storage units and logical structures for the set of storage units (page 3, Introduction: "XML documents are made up of *storage units* called entities, which contain either *parsed or unparsed data*. Parsed data is made up of characters, some of which form character data, and some of which form *markup*. *Markup encodes a description of the document's storage layout and logical structure*. XML provides a mechanism to impose constraints on the storage layout and logical structure.")

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to have combined W3C into McKindrick to include a reference to a specification of a particular transaction which has logical structures for storing at least one of definitions and references to documents as in W3C for the particular business transaction as in McKindrick since a reference is considered as a name or an identifier

and the transaction documents in McKindrick such as the purchase orders and the invoices, considered as the input and output documents, must have a document name for identifying purpose.

Regarding claim 8, which is dependent on claim 1, W3C discloses that the storage units comprise parsed data (page 3, Introduction: "XML documents are made up storage units called entities, which contain either parsed or unparsed data...").

Regarding claim 9, which is dependent on claim 1, McKindrick does not explicitly disclose the parsed data in at least one of the input and output documents comprises:

- character data encoding text characters in the one of the input and output document
- markup data identifying sets of storage units according to the logical structure of the one of the input and output documents

Instead McKindrick discloses the business transactions involved with manipulation and transfer data such as purchase orders and invoices where invoices are considered as the output documents produced from the data portion of the purchase orders, which are considered as the input document (pages 1-2).

W3C discloses that the parsed data comprises:

- character data encoding text characters in XML documents (page 3, Introduction: "*XML documents* are made up storage units ...*Parsed data* is made up characters, some of which form *character data* ..."; page 6, Characters: "A

parsed entity contains text, a sequence of characters, which may represent markup or character data

- markup data identifying sets of storage units according to the logical structure of XML documents (page 3, Introduction: "*XML documents* are made up storage units ... *Parsed data* is made up characters, some of which form character data, and some of which form *markup*. *Markup encodes a description of the document's storage layout and logical structure ...*")

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to have combined W3C into McKindrick since the XML business documents in McKindrick which function as input and output documents should comprise parsed data with claimed features since these features are characteristics of an XML document as taught in W3C.

Regarding claim 10, which is dependent on claim 9, W3C discloses that at least one of the sets of storage units encodes a plurality of text characters providing a natural language word (page 6, Document, page 7, Characters and page 8, Character Data and Markup: since the storage units encodes by character data and markup which are text, the storage units provide a natural language word).

Regarding claim 11, which is dependent on claim 8, W3C discloses that the interpretation information for at least one of the sets of storage units identified by a particular logical structure of at least one of the input and output documents, encodes respective definitions for sets of parsed characters (page 9: "the function of the markup

in an XML document is to describe its storage and logical structure and to associate attribute-value pairs with its logical structures. XML provides a mechanism, the document type declaration, to *define constraints on the logical structure* and to support the use of predefined storage units ... the XML document type declaration contains or points to markup declarations that provide a grammar for a class of documents. This grammar is known as a *document type definition, or DTD ...*").

Regarding claim 12, which is dependent on claim 8, W3C discloses that the storage units comprise unparsed data (page 3, Introduction: "XML documents are made up storage units called entities, which contain either parsed or unparsed data..." page 20, Physical Structures).

Regarding claim 13, which is dependent on claim 1, as mentioned in claims 4 and 5 above, it would have been obvious to one of ordinary skill in the art at the time of the invention was made to have modified McKindrick and W3C to include a repository in memory for storing all data related to the purchase transactions since it was well known in the art that any defined data for a program in a network should be stored in a memory of a server for using later on such as retrieving data, identifying data, or manipulating data.

Regarding claim 14, which is dependent on claim 13, W3C discloses that the repository of document types includes a document type for identifying participant process in the

network (page 9: "XML provides a mechanism, the document type declaration, to define constraints on the logical structure and to support the use of predefined storage units").

Regarding claim 15, which is dependent on claim 1, W3C discloses that the definitions of the input and output documents comprise document type definitions compliant with a standard Extensible Markup Language XML (page 9: "XML provides a mechanism, the document type declaration, to define constraints on the logical structure and to support the use of predefined storage units ... the XML document type declaration contains or points to markup declarations that provide a grammar for a class of documents. This grammar is known as a document type definition, DTD ... the DTD fro a document consists of both subsets taken together").

Regarding claim 16, which is dependent on claim 1, W3C discloses that the machine readable data structure including interpretation information comprises a document organized according to a document type definition compliant with a standard Extensible Markup Language XML (page 9: an XML document is a machine readable data structure organized according to a DTD compliant with the standard Extensible Markup Language).

Regarding independent claim 61, McKindrick does not disclose explicitly:

- defining a machine readable definition of an input document for a node in the network including resources to execute a process in the transaction, and a

machine readable definition of an output document for the node, the definitions the input and output documents comprising respective descriptions of sets of storage units and logical structures for the sets of storage units

- providing interpretation information for the logical structures to the node

Instead McKindrick discloses applying XML in financial area to provide better bank services and utilizing XML for on-line business transactions involved with manipulation and transfer of data in the Internet such as purchase orders, invoices, and customer information (pages 1-2). The purchase orders in McKindrick are considered as input documents, and the invoices are considered as output documents of the purchase orders in business transactions. Since the purchase orders as well as the invoices, which are the input and output documents, are in XML format, they definitely include information to provide the definition for said documents according to XML structures. And since the transaction documents are in XML format, these documents are machine-readable documents and should be stored in memory of a server accessible by at least one node in the network.

W3C discloses:

- defining a machine readable definitions of documents comprising respective descriptions of sets of storage units and logical structures for the sets of storage units (page 3, Introduction and page 9: XML documents are made up of storage units which contain either parsed or unparsed data where parsed data is made up characters some of which form character data, and some of which form

markup to encode a *description of the document storage layout and logical structures*).

- providing interpretation information for the logical structures (page 9: the function of the markup in an XML document is to associate *attribute-value* pairs with its logical structures)

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to have combined McKendrick into W3C for the following reason.

McKendrick discloses the transaction documents such as the purchase orders and the invoices in XML format for a business transaction over the Internet where a user can search and buy an item on-line and W3C discloses the structures of an XML document which comprises storage units and the logical structures for the set of storage units.

This motivates to combine W3C into McKendrick for supporting the business transaction documents in XML format using the XML characteristics disclosed in W3C.

Claims 62-71 are for a method of claims 2-5, 8-12, 15, and are rejected under the same rationale.

Regarding claim 72, which is dependent on claim 61, McKendrick and W3C do not disclose:

- providing a parser to generate event signals in response to logical structures in the definition of the input document

- providing event listener program which respond to the event signals to execute the process

Instead McKindrick discloses the Internet business transactions via purchase orders and invoices in XML format where the purchase orders and the invoices are considered as input documents and output documents (pages 1-2).

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to have modified McKindrick to include "providing a parser to generate event signals in response to logical structures..." and "providing event listener program which respond to the event signals to execute the process" for the following reason. The fact that McKindrick executes the transaction program by running the XML transaction documents which include logical structures suggests said parser and said event listener program as claimed, which are the must programs in the executing process.

Response to Arguments

7. Applicant's arguments filed 7/23/07 have been fully considered but they are not persuasive.

Applicants provide the new evidence to remove the reference McKendrick in the 103 rejection. However, McKendrick is maintained in the rejections in combination with W3C for the following reason.

The affidavit or declaration and exhibits must clearly explain which facts or data applicant is relying on to show completion of his or her invention prior to the particular date. Vague and general statements in broad terms about what the exhibits describe along with a general assertion that the exhibits describe a reduction to practice "amounts essentially to mere pleading, unsupported by proof or a showing of facts" and,

thus, does not satisfy the requirements of 37 CFR 1.131(b). In re Borkowski, 505 F.2d 713, 184 USPQ 29 (CCPA 1974). Applicant must give a clear explanation of the exhibits pointing out exactly what facts are established and relied on by applicant. 505 F.2d at 718-19, 184 USPQ at 33. See also In re Harry, 333 F.2d 920, 142 USPQ 164 (CCPA 1964) (Affidavit "asserts that facts exist but does not tell what they are or when they occurred.") (MPEP 715.07 [R-3]).

Applicants submitted the mapping of the language of the instant claims of the application to slide 30, one of Glushko's slides at his presentation on July 25, 1998.

This submission is not proper since:

- the claim mapping as required by the Board should be included in the declaration signed by inventors to show the possession of the inventors on the invented subject matter
- document used to make the claim mapping should be referred in the declaration
- in mapping, it needs to point out only which subject matter of the claims is mapped with a portion in the document that Applicants relied on, in this case, slide 30, *not what the Examiner relied on*

The Board notes that "McKendrick is considered as a printed publication published in the United States that qualifies as prior art under 35 U.S.C. 102 (a), and therefore may be antedated by a declaration that complies with the requirements of 37 C.F.R. 1.131 (b)" (DoA). The declaration does not comply with the requirements of 37 C.F.R. 1.131(b), and so does not predate the McKendrick reference.

Regarding new evidence of reduction to practice, Applicants shows that "CBL development was well under way prior to the May [sic] 11, 1998 date given in the declaration"(Remarks, page 14).

However, this merely shows that CBL had been developing until September 1998, when CBL 1.1 was released to public use at no cost.

In the article in December 1999 "How XML Enables Internet Trading Communities and Marketplaces" at <http://www.infoloom.com/gcaconfs/WEB/philadelphia99/glushko.HTM> by Dr. Glushko, it is true that work on CBL began in 1997. However, Dr. Glushko admitted that the early versions of CBL struggled with technical problems:

Work on CBL began in 1997, partly funded by a Department of Commerce's Advanced Technology Program research award on "Component-Based Commerce" to Veo Systems and three other firms [ATP]. Because of this research pedigree, early versions of CBL strove for logical completeness, expressiveness, and compactness to test the abstract modeling power of XML for electronic commerce and to identify requirements for development tools and runtime support.

This indicates that the first draft of CBL, which is the early version of CBL in 1997, did not worked and is not a completeness of the invention that Applicants relied on to show reduction to practice of the invention.

Applicants use the presentation by inventor Glushko on July 25, 1998 to show Glushko's actual reduction to practice two months before McKendrick.

The slides presented on that date are merely a PowerPoint document for presentation. They are not an evidence of a complete product that is guaranteed that it worked with testing.

*"In general, proof of actual reduction to practice requires a showing that the apparatus actually existed and worked for its intended purpose. However, "there are some devices so simple that a mere construction of them is all that is necessary to constitute reduction to practice." In re Asahi/America Inc., **>68 F.3d 442, 37 USPQ2d 1204, 1206< (Fed. Cir. 1995) (Citing Newkirk v. *>Lulejian<, 825 F.2d 1581, 3USPQ2d 1793 (Fed. Cir.1987) and Sachs v. Wadsworth, 48 F.2d 928, 929, 9 USPQ 252, 253 (CCPA 1931). The claimed restraint coupling held to be so simple a device that mere construction of it was sufficient to constitute reduction to practice. Photographs, coupled with articles and a technical report describing the coupling in detail were sufficient to show reduction to practice.)" (MPEP 715.07 III)"*

"For an actual reduction to practice, the invention must have been sufficiently tested to demonstrate that it will work for its intended purpose, but it need not be in a commercially satisfactory stage of development. If a device is so simple, and its purpose and efficacy so obvious, construction alone is sufficient to demonstrate workability. King Instrument Corp. v. Otari Corp., 767 F.2d 853, 860, 226 USPQ 402, 407 (Fed. Cir. 1985). For additional cases pertaining to the requirements necessary to establish actual reduction to practice see DSL Dynamic Sciences, Ltd. v. Union Switch & Signal, Inc., 928 F.2d 1122, 1126, 18 USPQ2d 1152, 1155 (Fed. Cir. 1991) ("events occurring after an alleged actual reduction to practice can call into question whether reduction to practice has in fact occurred"); Corona v. Dovan, 273 U.S. 692, 1928 C.D. 252 (1928) ("A process is reduced to practice when it is successfully performed. A machine is reduced to practice when it is assembled, adjusted and used. A manufacture [i.e., article of manufacture] is reduced to practice when it is completely manufactured. A composition of matter is reduced to practice when it is completely composed." 1928 C.D. at 262-263 (emphasis added).); Fitzgerald v. Arbib, 268 F.2d 763, 765-66, 122 USPQ 530, 531-32 (CCPA 1959) ("the reduction to practice of a three-dimensional design invention requires the production of an article embodying that design" in "other than a mere drawing")" (MPEP 2138.05).

It is noted that CBL 1.1 was released in September 1998 (Remarks, page 14), two months after Dr. Glushko's presentation July 25, 1998. The invention may be still under developing at the time of presentation.

Therefore, the second version of CBL, which is CBL 1.1, does not show reduction to practice prior the effective date of McKendrick reference.

It is noted that the next version of CBL, CBL 1.2 "was completed in November 1998" (Terry Allen, *Common Business Library (CBL)*, May 1999) after the filing date of the invention 10/16/98.

Also in the article of Dr. Glushko, CBL 2.0, which is the most successful version of CBL, was actually formed in January 1999, when Commerce One acquired Veo System:

*The acquisition of Veo Systems by Commerce One in January 1999 introduced to CBL a requirement for interoperability with EDI. CBL 2.0, in line with the lessons learned from CBL 1.0, aims for less abstraction, even if it means redundancy or less expressiveness, and greater compatibility with EDI standards and semantics. Using standard data element semantics provides a strong non-proprietary and interoperable semantic foundation for CBL, and gives companies using EDI today a clear migration path in CBL for transforming EDI applications to XML. CBL is freely available in repositories run by Commerce One as part of *marketsite.net*, as well as through those operated by *xml.org* and *Biztalk.org*.*

It is noted that January 1999 is after the filing date of the instant application. Therefore, according to the new evidence and related articles it appears that there is no actual reduction to practice prior September 1998. Accordingly, there was no reduction to practice prior the effective date of the McKendrick reference.

Only the filling of a US patent application which complies with the disclosure requirement of 35 USC 112 constitutes a constructive reduction to practice. A slide for presentation that is merely a written description, no matter how complete, has not been made the subject of a US patent application and does not qualify as reduction to practice.

Accordingly, Applicants have not established prior invention. The rejection is maintained.

Applicants provide the article *XML: Enabling Next-Generation Web Applications* to show where the passage quoted by McKendrick (Remarks, page 30) is included and to show that in the article, Microsoft teaches and promotes a Remote Procedure Call (RPC) use of XML and did not use an input document and an output document.

In response, it is noted that in the provided article, Remote Procedure Call (RPC) is mentioned and the business applications involving manipulation and transfer data such as purchase orders, invoices, and customer information using XML is also mentioned. The latter is what McKendrick quoted and used in his article wherein the purchase orders and the invoices are input documents and output documents. And that is the information that the Examiner would like to point out that XML is used to implement purchases orders and invoices in business applications.

Conclusion

8. This is a RCE of applicant's earlier Application No. 09/173,858. All claims are drawn to the same invention claimed in the earlier application and could have been finally rejected on the grounds and art of record in the next Office action if they had been entered in the earlier application. Accordingly, **THIS ACTION IS MADE FINAL** even though it is a first action in this case. See MPEP § 706.07(b). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no, however, event will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

9. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Wang et al. (US 6,490,217). Wang (US 5,917,913).

Fox et al. (US 6,560,581). Sato et al. (US 6,669,832).

Beckett et al. (US 6,724,896). Goldschmidt Iki et al. (US 2004/0078342).

Business Editors, *Sequoia Software to Launch First XML Transaction Server for Health Care*, Business Wire, Feb 13, 1998, ProQuest, pages 1-3.

Khare et al., XML: A Door to Automated Web Applications, IEEE 1997, pages. 78-87.

Cover, The XML Cover Pages Common Business Library (CBL), Google July 28, 1999, pages 1-3, <http://web.archive.org/web/20010603003905/xml.coverpages.org/cbl.html>.

Glushko, How XML Enalbles Internet Trading Communities and Marketplaces, Google, December 1999, pages 1-11,

<http://www.infoloom.com/gcaconfs/WEB/philadelphia99/glushko.HTM>.

Application/Control Number:
09/173,858
Art Unit: 2178

Page 22

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Cong-Lac Huynh whose telephone number is 571-272-4125. The examiner can normally be reached on Mon-Fri (8:30-6:00).

11. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Stephen Hong can be reached on 571-272-4124. The fax phone number for the organization where this application or proceeding is assigned is 571-273-4125.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Cong-Lac Huynh/
Cong-Lac Huynh
Primary Examiner
Art Unit 2178
1/14/08

R-292

CERTIFICATE OF ELECTRONIC FILING UNDER 37 CFR 1.8

I hereby certify that this correspondence is being electronically filed with the U.S. Patent & Trademark Office on 21 July 2008.

/Abby Berghella/
Abby Berghella

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Bart Alan MELTZER et al.
Application No.: 09/173,858
Confirmation No.: 4734
Filed: 16 October 1998
Title: Documents for Commerce in Trading
Partner Networks and Interface
Definitions Based on the Documents

Group Art Unit: 2178

Examiner: HUYNH, Cong Lac T.

CUSTOMER NO.: 22470

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RESPONSE TO FINAL OFFICE ACTION MAILED 23 JANUARY 2008

Sir:

In response to the Final Office Action mailed 23 January 2008, Applicants request entry of the following amendments and consideration of the following remarks. An appropriate request for extension of time accompanies this paper.

The Claims and their current status are reflected beginning on page 1.

Remarks begin on page 7.

TABLE OF CONTENTS

RESPONSE TO FINAL OFFICE ACTION MAILED 23 JANUARY 2008	i
Table of Exhibits	iii
New Exhibits to Inventor Declarations	iii
RCE Exhibits	iii
Appeal Exhibits.....	iv
RESTATEMENTS AND AMENDMENTS.....	1
In the Claims:.....	1
REMARKS.....	7
Summary	7
Dispositive New Evidence Proves Actual Reduction to Practice in January 1998.....	8
The Declarations Prove Development and Workability of Claimed Inventions	9
The “imdesc.xml” Exhibit Proves an Actual Reduction to Practice	10
The New Declarations Read Testimony and Exhibits on the Claims and Respond to the Examiner’s Arguments from the FOA	13
The New Evidence Clearly Removes McKendrick as a Reference	17
<i>Res Judicata</i> Does Not Apply when the Board Directs Consideration of New Evidence	17
The FOA Erroneously Excluded the RCE and Appeal Evidence, Contrary to Law and to the Board Directions	18
RCE Evidence Proved Actual Reduction to Practice on or Before July 25, 1998	18
RCE Evidence Corroborates the Pre-Appeal Declarations	20
RCE Evidence Establishes the 1998 Level of Skill in the Art as Immature, as Looking to These Inventors to Lead the Industry	22
CONCLUSION	24

Table of Exhibits

New Exhibits to Inventor Declarations¹

Submitted at this time:

- A. Tenenbaum, Jay M., Tripatinder S. Chowdhry and Kevin Hughes, "Eco System: An Internet Commerce Architecture" Computer May 1997: 48-55 (PDF 2)
- B. Glushko, Robert J., Jay M. Tenenbaum, Bart Meltzer, "An XML Framework for Agent-based E-commerce" Communications of the ACM, Vol. 42, No. 3, pp. 106-109 & 111-114 (March 1999) (PDF 12)
- C. "index.html" from cbl/072 directory (date stamped in 1997) (PDF 22)
- D. Selected files from cbl/072 directory (date stamped in 1997) (PDF 29)
- E. Selected files from cbl/075 directory (date stamped before January 21, 1998) (PDF 77)
- F. "Requirements and Tasks for the January Demo, (Updated 1/6/98 by Kenneth)", file named "demo_req_tasks.html" from Veo/web/dev/documents/old/demo directory (date stamped before January 21, 1997) (PDF 171)
- G. "imdesc.xml" from cbl/ingram/01 directory (date stamped before January 21, 1997) (PDF 179)
- H. Selected files from cbl/ingram/01 directory (date stamped before January 21, 1997) (PDF 182)
- I. Glushko, Robert J., Implementing Domain-specific Commerce Languages with a Common Business Library (delivered July 25, 1998) accessed at <http://groups.haas.berkeley.edu/citm/conferences/cec/Presentations/Session3/glushko.pdf> on October 26, 2006 (PDF 249)
- J. Excerpts from W3C "note" WSDL version 1.1 (March 15, 2001) accessed at <http://www.w3.org/TR/wsdl> (PDF 282)

RCE Exhibits

Submitted after appeal:

- [K.] Glushko, LaPlante et al., Document Standards & Technologies for Commerce Applications (delivered Sept. 1, 1998) accessed at

¹ These Exhibits are collected in a single PDF file. If the PDF file is delivered in its native format, it will include bookmarks. Otherwise, the page numbers within the 286 page PDF file can be used for quick navigation.

http://www.google.com/search?q=cache:hq5pefHRwksJ:seminars.seyboldreports.com/1998_san_francisco/ETAPE_26.html+%22cbl+1.0%22+veo&hl=en&gl=us&ct=clnk&cd=7 on Oct. 26, 2006, 28 pp.

[L.] xCBL.org, About xCBL (copyright 2000) accessed at <http://www.xcbl.org/about.shtml>, on July 20, 2007, 2 pp.

[M.] Allen, Common Business Library (CBL) (May 1999) accessed at <http://www.infoloom.com/gcaconfs/WEB/granada99/all.HTM> on July 20, 2007, 9 pp.

[N.] Bosak, UBL Update, OASIS Symposium on the Future of XML Vocabularies, Slide 3 (Apr. 25, 2005) viewed at www.oasis-open.org/events/symposium_2005/slides/bosak.pdf on July 20, 2007, 10 pp.

[O.] Meltzer and Glushko, XML and Electronic Commerce: Enabling the Network Economy, SIGMOND Record, Vol. 27, No. 4, 21-24 (Dec. 1998), 4 pp.

[P.] Registries and Repositories – XML/SGML Name Registration (about Nov. 1998) accessed at <http://xml.coverpages.org/registryColl.html> on October 26, 2006, 30 pp.

[Q.] Microsoft Corp., XML: Enabling Next-Generation Web Applications (Apr. 3, 1998) accessed at <http://msdn.microsoft.com/archive/default.asp?url=/archive/en-us/dnarxml/html/xmlwp2.asp> on July 21, 2007, 15 pp.

[R.] Bosworth, General Manager, Microsoft Corp., Europe '98, Microsoft's Vision for XML (May 18-21, 1998) viewed at <http://xml.coverpages.org/bosworthXML98.html> on July 21, 2007, 10 pp.

[S.] Winer, XML-RPC for Newbies (July 14, 1998) viewed at <http://www.scripting.com/davenet/1998/07/14/xmlRpcForNebies.html> on July 21, 2007, 6 pp.

[T.] Walsh, Microsoft spearheads protocol push, InfoWorld Electric (July 10, 1998) viewed at <http://infoworld.com/cgi-bin/displayStory.pl?98071.whsoap.htm> on July 21, 2007, 2pp.

[U.] Merrick et al., US 7,028,312, XML Remote Procedure Call (XML-RPC)

Appeal Exhibits

Submitted during appeal and resubmitted after appeal:

[J, above] Glushko, Implementing Domain-specific Commerce Languages with a Common Business Library (delivered July 25, 1998) 32 pp.

[B, above] Glushko et al., An XML Framework for Agent Based E-Commerce, Communications of the ACM, Vol. 42, No. 3, pp. 106-114 (Mar. 1999), 9 pp.

[V.] Glushko and McGrath, Document Engineering: Analyzing and Designing Documents for Business Informatics and Web Services (MIT Press 2005), 2 pp. excerpt

[W.] Sall, Kenneth B., XML Family of Specifications: A Practical Guide (Addison Wesley 2002), 1 pp.

RESTATEMENTS AND AMENDMENTS

In the Claims:

The following is a list of claims currently pending in this application and their current status. This listing of claims replaces all prior versions and listings in this application.

1. (Previously presented) An interface for transactions among nodes in a network including a plurality of nodes which execute processes involved in the transactions, the interface being stored in a computer readable medium, comprising:

a machine readable specification of an interface to transaction processes stored in memory accessible by at least one node in the network, including interpretation information providing a definition of an input document, and a definition of an output document, the definitions of the input and output documents comprising respective descriptions of sets of storage units and logical structures for the sets of storage units.

2. (Original) The interface of claim 1, wherein the interpretation information includes data type specifications for at least one logical structure in the definitions of the input and output documents.

3. (Original) The interface of claim 1, wherein the interpretation information includes at least one data structure mapping predefined sets of storage units for a particular logical structure in the definitions of the input and output documents, to respective entries in a list.

4. (Original) The interface of claim 1, including a repository in memory accessible by at least one node in the network storing a library of logical structures, and

interpretation information for logic structures.

5. (Original) The interface of claim 1, wherein the machine readable specification includes a document compliant with a definition of an interface document including logical structures for storing an identifier of a particular transaction, and at least one of definitions and references to definitions of input and output documents for the particular transaction.

6. (Original) The interface of claim 1, wherein the machine readable specification includes a document compliant with a definition of an interface document including logical structures for storing an identifier of the interface, and for storing at least one of specifications and references to specifications of a set of one or more transactions supported by the interface.

7. (Original) The interface of claim 6, wherein the machine readable specification includes a reference to a specification of a particular transaction, and the specification of the particular transaction includes a document including logical structures for storing at least one of definitions and references to definitions of input and output documents for the particular transaction.

8. (Original) The interface of claim 1, wherein the storage units comprise parsed data.

9. (Original) The interface of claim 8, wherein the parsed data in at least one of the input and output documents comprises:

character data encoding text characters in the one of the input and output documents, and

markup data identifying sets of storage units according to the logical structure of the one of the input and output documents.

10. (Original) The interface of claim 9, wherein at least one of the sets of storage units encodes a plurality of text characters providing a natural language word.

11. (Original) The interface of claim 8, wherein the interpretation information for at least one of the sets of storage units identified by a particular logical structure of at least one of the input and output documents, encodes respective definitions for sets of parsed characters.

12. (Original) The interface of claim 8, wherein the storage units comprise unparsed data.

13. (Original) The interface of claim 1, including a repository stored in memory accessible by at least one node in the network of document types for use in a plurality of transactions, and wherein the definition of one of the input and output documents includes a reference to a document type in the repository.

14. (Original) The method of claim 13, wherein the repository of document types includes a document type for identifying participant processes in the network.

15. (Original) The interface of claim 1, wherein the definitions of the input and output documents comprise document type definitions compliant with a standard Extensible Markup Language XML.

16. (Original) The interface of claim 1, wherein the machine readable data structure including interpretation information comprises a document organized according to a document type definition compliant with a standard Extensible Markup Language XML.

17. – 60. (Cancelled).

61. (Original) A method for programming a commercial transaction in a network, comprising:

defining a machine readable definition of an input document for a node in

the network including resources to execute a process in the transaction, and a machine readable definition of an output document for the node, the definitions of the input and output documents comprising respective descriptions of sets of storage units and logical structures for the sets of storage units; and

providing interpretation information for the logical structures to the node.

62. (Original) The method of claim 61, wherein the interpretation information includes data type specifications for at least one logical structure in the definitions of the input and output documents.

63. (Original) The method of claim 61, wherein the interpretation information includes at least one data structure mapping predefined sets of storage units for a particular logical structure in the definitions of the input and output documents, to respective entries in a list.

64. (Original) The method of claim 61, the step of providing interpretation information includes providing a repository in memory accessible by at least one node in the network storing a library of logical structures, and interpretation information for logic structures.

65. (Original) The method of claim 61, including defining a machine readable specification of an interface including a document compliant with a definition of an interface document including logical structures for storing an identifier of a particular transaction, and at least one of the definitions and references to the definitions of the input and output document.

66. (Original) The method of claim 61, wherein the storage units comprise parsed data.

67. (Original) The method of claim 66, wherein the parsed data in at least one of the

input and output documents comprises:

character data encoding text characters in the one of the input and output documents, and

markup data identifying sets of storage units according to the logical structure of the one of the input and output documents.

68. (Original) The method of claim 67, wherein at least one of the sets of storage units encodes a plurality of text characters providing a natural language word.

69. (Original) The method of claim 67, wherein the interpretation information for at least one of the sets of storage units identified by a particular logical structure of at least one of the input and output documents, encodes respective definitions for sets of parsed characters.

70. (Original) The method of claim 66, wherein the storage units comprise unparsed data.

71. (Original) The method of claim 61, wherein the definitions of the input and output documents comprise document type definitions compliant with a standard Extensible Markup Language XML.

72. (Original) The method of claim 61, including:

providing a parser to generate event signals in response to logical structures in the definition of the input document; and

providing event listener programs which respond to the event signals to execute the process.

73. (New) An interface for transactions among nodes in a network including a plurality of nodes which execute processes involved in the transactions, the interface being stored in a computer readable medium, comprising:

a machine readable specification of an interface to an operation stored in memory accessible by at least one node in the network, including interpretation information providing a definition of an input document, and a definition of an output document, the definitions of the input and output documents comprising respective descriptions of sets of storage units and logical structures for the sets of storage units.

74. (New) The interface of claim 73, wherein the interpretation information includes data type specifications for at least one logical structure in the definitions of the input and output documents.

REMARKS

Claims 1-16 and 61-72 have been examined in this application. Claims 17-60 were cancelled long ago.

Claims 73 and 74 are new. They are directed to specification of an interface to an operation, rather than an interface to plural transaction processes. Support for the "operation" nomenclature is found in the application at 17, 25-27, and 86. Applicants do not intend to introduce any new matter with these claims.

Summary

This submission presents five new declarations from inventors and a non-inventor and approximately 300 pages of corroborating exhibits, based on archives that were discovered after appeal (collectively, the "new evidence"). The new corroborating exhibits include internal memoranda and computer program listings. The new declarations prove actual reduction to practice before January 21, 1998 and address the claims. This actual reduction to practice predates the McKendrick reference by eight months, thereby putting the claims in condition for allowance.

Our second thrust is that declarations are not an exclusive means for establishing, pursuant to 35 USC section 102, that the inventors had an actual, publically demonstrated and reported reduction to practice of the claimed data structures before McKendrick was published. The Office's rules (e.g., current rule 131 and predecessor rule 75) describe a safe harbor for showing actual reduction to practice prior to the date of a reference, but the rules do not and cannot limit the statute or limit the alternative ways of proving actual reduction to practice. *Ex Parte Foster*, 105 O.G. 261 (Comm'r Pat. 1903). Evidence other than declarations (the "previous evidence") was submitted, but given no weight, due to the Examiner's mistaken invocation of *res judicata*. *Res judicata* cannot be applied post-appeal to exclude evidence from consideration which the Board effectively directed the Examiner to consider. See, *On Request for Rehearing*, at 7 (May 21, 2007); see, *Decision of Petition*, at 2 (Sept. 17, 2007) (quoting May 21 decision and denying petition as moot because RCE already filed with new evidence). The previous evidence, standing alone or corroborating the pre-appeal declarations, removes McKendrick as a reference.

Our third thrust is, again, that the 33 words of McKendrick on which the Examiner relies need to be evaluated from the perspective of one of ordinary skill in the art, after considering evidence regarding the 1998 level of skill in the art. *Graham v. John Deere*, 383 U.S.1, 17-18 (1966); *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S.#__, 127 S. Ct. 1727, 82 U.S.P.Q.2d 1385 (2007); *Ex Parte Jud*, 82 U.S.P.Q.2d 1280, 2007 Pat. App. LEXIS 9 (BPAI 2007). The exhibits of record demonstrate that these inventors worked hard in 1998 and 1999 to advance the level of skill in the art. They were at the leading edge of technology, promoting something that now seems familiar. Their actual reduction to practice came before and their evangelism closely followed publication of the XML 1.0 recommendation on February 10, 1998. Deliberate effort is required to avoid applying hindsight to an article published in 1998 when ordinary programmers, tutored by these inventors, were trying to figure out what XML was, what tools and strategies Microsoft was promoting, and what the emerging alternatives offered. Thus far, neither the Examiner nor the Board has given any weight or expression to the 1998 level of skill in the art. As a result, McKendrick has been misinterpreted and misapplied to these claims.

For these three reasons, all previous rejections should be withdrawn, placing the claims in condition for allowance.

Dispositive New Evidence Proves Actual Reduction to Practice in January 1998

We are submitting dispositive new evidence based on old corroborating computer files that were first provided to counsel after appeal. Former Veo employee Kevin Hughes happened to have an old computer on which copies of internal memoranda and programs from 1997-1998 were found. This historical material was analyzed, excerpted into exhibits and provided to the witnesses. The witnesses whose declarations we now submit include four out of five inventors and Kevin Hughes².

² A petition to the Office of Petitions is being submitted at the same time, under the revised directive of MPEP § 715.04(I) at 700-281 (spanning left and right columns). All of the inventors except one signed the accompanying declarations. The missing inventor's physical address was located using property tax records. Packages were sent via Express Mail to both the missing inventor's PO Box and his physical address. Both packages were refused by inventor Terry Allen, with the notation "REF T.A." and returned by the USPS to counsel. This is a more elaborate procedure and complete proof than was necessary to satisfy the Board on appeal. Accordingly, the Office of Petitions is expected to direct that the remaining inventors' declarations be given the same weight as if all inventors had signed.

Alternatively, a non-inventor's declaration is submitted from a percipient witness, as authorized by the

The Declarations Prove Development and Workability of Claimed Inventions

The declarations follow a consistent structure. Personal information appears in paragraphs 1-5, which varies by witness. For Kevin Hughes, this personal information includes authentication of memoranda and program listings that were reprinted from his computer. The declarations discuss Veo's shift from attempts to use CORBA technology in 1997 to development of a platform that used XML in 1997-1998, before W3C published the XML 1.0 recommendation in February 1998. See, ¶¶ 6-7. Veo's development using XML responded to a long felt need for an object-oriented architectural framework for Internet commerce. *Id.*

The inventors developed and tested their technology for document exchanges and interface definitions using XML. They correctly understood the workability of their technology and its vast superiority to CORBA data object exchanges and CORBA data definitions, before W3C published the XML 1.0 recommendation. See, ¶¶ 7-8. Two data structures defining transaction interfaces that used input and output documents, dated January 3, 1998 and July 25, 1998, are reproduced in the declaration and discussed. See, ¶¶ 9-10. Each of these data structures is evidence of an actual reduction to practice, because the reprinted data structures were stored in computer memory, run on computers, tested, and accepted by the inventors as workable for their intended purpose. The witnesses identify and describe the corroborating exhibits, which include listings of program code from at least development versions 0.7.2, 0.7.5 and Ingram /01, all of which precede January 21, 1998. See, ¶¶ 11-15.

The declarations emphasize preparation for a demonstration to important potential customer Ingram Micro and, particularly, a computer program file named "imdesc.xml" taken from version development Ingram /01. See, ¶¶ 14-16. The demonstration to Ingram Micro was important because of a potential strategic relationship. Requirements and activities in preparation for the demonstration are

statute § 102 and *Ex parte Foster*, 105 O.G. 261 (Comm'r Pat. 1903) (copy previously submitted). We point out that MPEP § 715.04 cites *Ex Parte Foster* as good authority, despite its early date. The statute only disqualifies claims from being granted if a reference predates the inventors' work. The statute does not limit the Applicants' proof to the safe harbor of rule 131 or to any particular evidence. The case *Ex Parte Foster* makes it clear that rule 131 is not an exclusive means of presenting evidence regarding inventors' work that predates a reference. The MPEP continues to endorse *Ex Parte Foster*, even though it was decided 105 years ago. A colleague of the inventors who retained a copy of early development code is a natural witness.

detailed in a memorandum. (Ex. F) Correlated with the memorandum, the computer program listings include modules with names that match every module listed as a requirement for the demonstration. (Exs. D, E, G, and H) The module "imdesc.xml" is referred to and explained in the demonstration preparation memorandum. (Ex. F) The data structure in "imdesc.xml" is an actual reduction to practice of a data structure within the scope of these claims, which the inventors used in methods.

The declarations refer to work three years later by members of W3C that promotes the use of data structures to define Web service interfaces by reference to input and output documents. See, ¶ 18. The 2001 W3C data structures are essentially the same as those that Veo reduced to practice before January 21, 1998. This is proof either of copying or industry acceptance three years later of Veo's designs.

Finally, the declarations refute the Examiner's speculation that CBL was not tested or demonstrated to be a workable approach before the release of CBL version 1.1 or 2.0. The witnesses testify that CBL worked for many versions and many months prior to the September 1998 release of version 1.1. See ¶ 20.

The accompanying exhibits include Veo internal memoranda (Exs. C & F), selected listings of computer programs that were written and tested on Veo's computers (Exs. D, E, G, H), and articles (Exs. A, B, I & J). For the program listings, a relative path name of the computer disk memory file directory from which the listing was printed is given in the exhibit list and reiterated in the declarations.

The declaration testimony proves existence and sufficient testing of a data structure and method within the scope of these claims to qualify as an actual reduction to practice. The date proven for the actual reduction to practice is before January 21, 1998, the scheduled date for demonstration of a system using the data structure. The exhibits reveal details of the data structure.

The "imdesc.xml" Exhibit Proves an Actual Reduction to Practice

The declaration testimony regarding the file "imdesc.xml" (Ex. G) is dispositive. Testing and use of the data structure found in "imdesc.xml" is described in both the declarations (¶¶ 13-16) and an internal memorandum. (Ex. F at PDF 175) The internal memorandum (Ex. F at PDS 175-76) lists program modules required for the demonstration. Copies of these programs are provided in exhibits D, E, G, and H. The

internal memorandum concludes by giving January 21, 1998 as the planned date for demonstration of the system to Ingram Micro. (Ex. F at PDF 178)

The imdesc.xml file dated "2 Jan 1998" (Ex. G) provides details of functional data that defines an interface including input and output XML documents, per the independent claims and the dependent claims that call for use of XML. The interface definition in this file references so-called ".dtd" schema files that contain full definitions of the input and output XML documents. Copies of the "order.dtd" and "invoiceo.dtd" schema files are provided. (Ex. E, order.dtd at 38 / PDF 115; invoiceo.dtd at 29 / PDF 106)

The imdesc.xml file dated "2 Jan 1998" (Ex. G), when placed in memory and used in a test or demonstration, is an actual reduction to practice of a data structure and method. The listings of imdesc.xml and related computer programs were printed from a computer disk. The listed files have been stored in computer readable memories continuously for more than 10 years, since they were created in 1997-98. The witnesses tested and used these programs, because that was the core of their business.

One of skill in the art would see that these inventors had full possession of their interface definition technology on or before January 2, 1998, based on the progression of the data structures from Ex. G to Ex. J and on to page 45 of the original application. First, an excerpt from the computer program listing for a demonstration to Ingram Micro (hence "im"desc.xml), Ex. G at PDF 180 (Jan 2, 1998):

```
<service.set>
<service>
  <service.name>Ordering and Fulfillment
</service.name>
  <service.function.sequence>
    <service.function>
      <doctype from.party="any" to.party="ingram">order.dtd</doctype>
      <doctype from.party="ingram" to.party="any">ack.dtd</doctype>
    </service.function>
    <service.function>
      <doctype from.party="ingram" to.party="any">invoiceo.dtd</doctype>
      <doctype from.party="any" to.party="ingram">ack.dtd</doctype>
    </service.function>
    <service.function>
      <doctype from.party="ingram" to.party="any">shipnote.dtd</doctype>
    </service.function>
    <service.function>
      <doctype from.party="any" to.party="ingram">paynoteo.dtd</doctype>
      <doctype from.party="ingram" to.party="any">ack.dtd</doctype>
    </service.function>
  </service.function.sequence>
</service>
```

Then, from Glushko's presentation at a conference, Ex. I at PDF 279 (July 25, 1998):

```
<service>
<service.name>Order Service</service.name>
<service.location>www.veosystems.com/order</service.location>
<service.op>
  <service.op.name>Submit Order</service.op.name>
  <service.op.inputdoc>po.dtd</service.op.inputdoc>
  <service.op.outputdoc>poack.dtd</service.op.outputdoc>
</service.op>
< service.op>
  < service.op.name>Track Order</service.op.name>
  <service.op.inputdoc>request.track.dtd<service.op.inputdoc>
  <service.op.outputdoc>response.track.dtd<service.op.outputdoc>
</service.op>
</service>
```

Finally, from the application, page 45 (Oct. 16, 1998):

```
<service>
<service.name>Order Service</service.name>
<service.location>www.veosystems.com/order</service.location>
<service.op>
<service.op.name>Submit Order</service.op.name>
<service.op.inputdoc>www.commerce.net/po.dtd</service.op.inputdoc>
<service.op.outputdoc>
  www.veosystems.com/invoice.dtd</service.op.outputdoc>
</service.op>
< service.op>
< service.op.name>Track Order</service.op.name>
<service.op.inputdoc> www.commerce.net
  /request.track.dtd<service.op.inputdoc>
<service.op.outputdoc>
  www.veosystems.com/response.track.dtd<service.op.outputdoc>
</service.op>
</service>
```

These three program fragments include functional data that defines interfaces to multiple functions, a.k.a. operations, which were useful for purchases and purchase fulfillment. The first two fragments illustrate a more elaborate document exchange than

the fragment from the patent application, because they include acknowledgements of input documents.

From January 2 to July 25, 1998, both dates prior to McKendrick, the service name changed from "Ordering and Fulfillment" to "Order Service". The input document name changed from "order.dtd" to "po.dtd". The definition of one output document changed for "ack.dtd" to "poack.dtd". In the October 16, 1998 application, corresponding elements retained the names "Order Service" and "po.dtd". The variations among the January, July and October data structures evince a common approach and three examples of data structures within the general scope of the claims. This proves early possession of working programs within the general scope of these claims, that were an actual reduction to practice and, therefore, remove McKendrick as a reference.

The New Declarations Read Testimony and Exhibits on the Claims and Respond to the Examiner's Arguments from the FOA

The declarations go further than the narrative above in mapping the actual reduction to practice to the claims. In paragraph 19, the declarations include subparagraphs labeled to match individual claims. The testimony reads on the claims and references corroborating exhibits. Rather than repeating the whole text of the declarations, which we trust the Examiner to study, we reiterate here only the testimony that reads on the independent claims and a sample set of dependent claims.

Claim 1: In Exhibits G-I, both imdesc.xml and Slide 30 depict machine-readable interface specifications. The imdesc.xml file (Ex. G) typically is found in a file directory on a machine-readable storage media. It is functional data that defines an interface by reference to input and output documents. The PowerPoint Slide 30 (Ex. I) was taken from a computer file similar to imdesc.xml and pasted into the presentation. The archive for imdesc.xml and the PowerPoint presentation both were stored on machine-readable storage media. Both data structures define an interface to a transaction process. The imdesc.xml defines an interface with several service "functions". Slide 30 defines multiple service "operations". In imdesc.xml, one of the input documents is an "order"; in Slide 30, there is a "po", which is short for purchase order. In both transaction interface definitions, an acknowledgement is sent, an "ack" and a "poack", respectively. The input document definitions are referenced by "order.dtd", "invoiceo.dtd", "paynoteo.dtd", "po.dtd" and "request.track.dtd", which are data type definition (dtd) files. The output

document definitions are referenced by “ack.dtd”, “poack.dtd” and “response.track.dtd”. The January 1998 demonstration scenario (Ex. F, quoted in the declarations) makes it clear that these interface definitions were published to nodes on a network that might desire to invoke the functions for which interfaces were defined. The demonstration scenario (Ex. F) and slides (Ex. I) make it clear that this interface was hosted and accessible to a plurality of nodes on a network.

The mapping above is taken from the accompanying declarations. ¶ 19, claim 1. The corroborating computer program listings from January and July, 1998 (Exs. G and I), predate McKendrick. This testimonial and documentary evidence is consistent with the previously submitted declarations and establishes a date of actual reduction to practice before January 21, 1998.

Responding to the Examiner’s view that written descriptions fail to qualify as constructive reductions to practice (FOA at 19), we point out that historical computer program listings are good evidence that a program ran on a computer, which is an actual reduction to practice, not a constructive reduction. Only museums keep ten year old computers in service and we could not submit an old computer via the PTO’s Electronic Filing System if we tried. So, it makes sense for us to submit computer program listings and testimony that the computer programs ran on computers. The computer program listings are more than adequate corroboration for the declaration testimony of an actual reduction to practice.

Responding to the Examiner’s conjecture (FOA at 17-19) that CBL was not recognized as satisfactory for its intended purpose until version 1.1, released in September 1998, or version 2.0 released in January 1999, we provide the inventors’ contrary testimony. As addressed below, the Examiner’s argument about when CBL worked is not supported by the quoted passages. The inventors’ declarations, in their final paragraph (¶ 20) make it clear that “CBL worked for its intended purpose for many months and many versions prior to the September 1998 release of public version 1.1.” Since we have testimonial evidence on point, the Examiner’s conjecture must be set aside. Moreover, code samples of CBL from development versions 0.7.2, 0.7.5 and Ingram /01 accompany the declarations and predate January 21, 1998. The declarations give clear testimony that these code samples were understood to work as intended, after testing and preparation for a demonstration to Ingram Micro that was

scheduled for January 21, 1998. The record now includes working versions of CBL and related programs from December 1997 and January 1998.

The Examiner's requirement (FOA at 17, bottom) that Applicants need "evidence of a complete product that is guaranteed that it worked with testing," misstates the applicable rule of law. Applicants need not have completed a commercial product and need not have guaranteed that it worked based on extensive testing. "[I]n order for there to be a reduction to practice, there is no requirement that the invention when tested be in a commercially satisfactory stage of development." *King Instrument v. Otari Corp.*, 767 F.2d 853, 861, 226 U.S.P.Q. 402 (Fed. Cir. 1985) *cert. denied* 475 U.S. 1016 (1986). "Some devices are so simple and their purpose and efficacy so obvious that their complete construction is sufficient to demonstrate their workability." *Mahukar v. C.R. Bard, Inc.*, 79 F.3d 1572, 1578, 38 U.S.P.Q.2D 1288 (Fed. Cir. 1996) (*citing King Instrument*). A demonstration prototype is enough, such as the prototype prepared for demonstration to Ingram Micro. The witnesses have testified that they tested the prototype sufficiently to appreciate that it was workable. While preparation for the important demonstration to Ingram Micro manifestly involved substantial effort and testing, there is no evidence of record to suggest that the elegant data structure reproduced above would have required any testing to show that it would work for its intended purpose. Three years later, a W3C note regarding WSDL version 1.1 (Ex. L at PDF 285-86) confirmed that the inventors were correct when they understood that their approach worked for its intended purpose – the note promotes use of virtually the same data structure as in "indesc.xml". While we have not attempted to prove that Veo had a completed product in January 1998, we have proven testing and appreciation that the inventive data structure would work, which is enough to remove McKendrick as a reference when the correct legal standard is applied.

The Board's decision *On Request for Rehearing*, at 3 (May 21, 2007), quotes *In re Stempel*, 241 F.2d 755, 759-60, 113 U.S.P.Q. 77, 81 (CCPA 1957) as only requiring Applicants to show "priority with respect to so much of the claimed invention as the [McKendrick] reference happens to show." The declarations and exhibits make a much more complete showing than McKendrick of a computer program in memory with a data structure that defined a transaction interface as including input and output documents. *Discussed, infra*, at 19. We have proven that Veo reduced to practice much more than

McKendrick describes, as *In re Stempel* allows us to, months before McKendrick was published.

When the new evidence related to claim 1 is considered and the proper legal standard is applied, the Examiner should find that Applicants successfully have sworn behind McKendrick and removed it as a reference. Therefore, claim 1 is in condition for allowance.

Claims 8-12: The input and output documents in both “imdesc.xml” (Ex. G) and Slide 30 (Ex. I) are defined as XML documents. Generally, XML documents compliant with the February 1998 recommendation can be parsed or unparsed data. XML documents may encode text characters and the text characters may provide a natural language word. XML documents generally include markup data to identify sets of storage units. The data structures considered with an understanding of the XML standard show that the inventors’ working programs incorporated the features of claims 8-12, thereby removing McKendrick as a reference.

Responding to the Examiner’s argument (FOA at 16) regarding claim mapping, the new declarations are mapped to both independent and dependent claims. However, there is no legal requirement that this be done in the declarations. The Examiner does not cite any case or refer to any MPEP section. The cases cited by the Board give Applicants the option of explaining documentary evidence either in their argument (*Decision on Appeal*, at 7 (Aug. 31, 2006)) or in the declarations. *Id.*, at 9. The Examiner’s requirement for the mapping to be in the declarations goes beyond what the case law requires.

We have gone beyond what the case law or the Board requires to remove McKendrick as a reference against the dependent claims. This paper reiterates the evidence that reads the actual reduction to practice on dependent claims 8-12. The declarations are similarly mapped to the remaining dependent claims and incorporated herein by reference. Therefore, the dependent claims are in condition for allowance.

Claim 61: Independent method claim 61 is a corollary to the data structure of claim 1. The evidence described in the context of claim 1 generally applies to claim 61. In addition, the declarations show that, in the course of creating “imdesc.xml” (Ex. G) and Slide 30 (Ex. I), the authors of those documents went through the process of defining machine readable interface definitions including an input and output document.

Veo's transaction interface definition data structures were provided to network nodes that requested them. This is sufficient evidence, viewed in light of the extensive exhibits provided, of actual reduction to practice of the method of claim 61.

We see no additional arguments in the FOA that the Examiner might apply particularly to claim 61, beyond the arguments to which we responded above. Therefore, claim 61 is in condition for allowance.

Other claims: The declarations are similarly direct about reading Veo's actual reduction to practice on the remaining claims.

The New Evidence Clearly Removes McKendrick as a Reference

Therefore, the declarations and extensive exhibits remove McKendrick as a reference and put the claims in condition for allowance.

Res Judicata Does Not Apply when the Board Directs Consideration of New Evidence

The Examiner misapplied *res judicata* (FOA at 2-3) and appears to have used it as a reason not to consider the RCE evidence. The Examiner correctly referred to MPEP § 706.03(w), but missed the predominance of cases listed in the right hand column in which "*res judicata* rejections were reversed." The MPEP reports that new evidence was not subject to *res judicata* and that rejections on that basis were reversed in the cases of *In re Herr*, 377 F.2d 610, 153 USPQ 548 (CCPA 1967); *In re Russell*, 439 F.2d 1228, 169 USPQ 426 (CCPA 1971); and *In re Ackermann*, 444 F.2d 1172, 170 USPQ 340 (CCPA 1971). More recently, the Federal Circuit has reaffirmed that new evidence cannot be excluded by a *res judicata* rejection. *In re Donohue*, 766 F.2d 531, 533, 226 U.S.P.Q. 619 (Fed. Cir. 1985).

The legal principle of *res judicata* would bar any consideration of new evidence, if it applied. The Board and the Chief Administrative Judge expressly invited Applicants to submit evidence to the Examiner. "We note that if Appellants wish to have the newly presented evidence considered by the Examiner, the proper procedure is to file a Request for Continued Examination (RCE) under 37 C.F.R. § 1.114."³ Application of *res judicata* to exclude evidence is wholly inconsistent with the Board's rulings.

³ On *Request for Rehearing* at 7 (May 21, 2007); see, *Decision of Petition*, at 2 (Sept. 17, 2007) (quoting May 21 decision and denying petition because RCE already filed with new evidence).

It was an error for the Examiner to raise *res judicata* as barring consideration of evidence referred by the Board back to the Examiner and additional evidence presented with the RCE. The excluded RCE and appeal evidence proves the inventors' possession of their claimed invention prior to McKendrick, and shows how one of ordinary skill in the art would have understood McKendrick. Therefore, it was prejudicial error to exclude this evidence from consideration.

The FOA Erroneously Excluded the RCE and Appeal Evidence, Contrary to Law and to the Board Directions

The Board sustained rejection of claims 1-16, 61-72 under 35 U.S.C. § 103(a) as unpatentable over McKendrick, *Banks begin to play with XML*, Bank Technology News, Sep 1998, Vol. 11, Iss. 9, pg. 6, 2 pgs, in view of W3C, *Extensible Markup Language (XML) 1.0*, 2/10/98, pages 1-37. We responded (*Resp.* at 7 *et seq.*) by submitting evidence for the Examiner to consider *de novo*, as the Board directed.

RCE Evidence Proved Actual Reduction to Practice on or Before July 25, 1998

The Examiner erred in failing to consider clear evidence of a public demonstration that Veo had reduced its inventive data structure to practice two months before McKendrick published. The first set of RCE evidence, which the Examiner erroneously disregarded, included Glushko's Slides (Ex. I), 1999 Article (Ex. B) and 2005 Book (Ex. V). (*Resp.* at 8-11 & 15-18) The Examiner excluded the Glushko evidence as improperly submitted (FOA at 16), even though the Board reacted to the same evidence by directing us to have the Examiner consider it.

Dr. Glushko revealed an inventive data structure at a conference on July 25, 1998. This is documented in public reports of the conference. (Ex. I; *Resp.* at 249) Public reports of the inventive data structure and its use in demonstration projects are proof under 35 USC section 102 (*Ex Parte Foster*) that Veo and Dr. Glushko were in possession of the claimed invention two months before publication of McKendrick. The elegant data structure revealed at the conference, when placed in memory, self-evidentially worked as intended. Dr. Glushko did not find it necessary to do more than present the data structure on a PowerPoint slide in order to demonstrate its workability to the audience.

The Examiner gives bullet point arguments why the “submission is not proper”, but does not offer any supporting case law, PTO rule or MPEP passage. The Examiner has no legal basis for refusal to consider this non-declaration evidence. Because the Examiner argued that the submission was improper, she did not consider or criticize the evidence and did not challenge the conclusions drawn from it. There are no arguments regarding the quality of the evidence that call for a response, only procedural arguments.

Responding to the Examiner’s procedural arguments (FOA at 16) that claim mapping and reference to documentary evidence “should” be included in the declarations, “should” does not mean “must.” The cases cited by the Board in the appeal decision make it clear that the Applicants have the option of explaining the evidence either in their briefs or in declarations. *Decision on Appeal*, at 7 & 9.

The Examiner’s next argument (FOA at 16) is that mapping needs only relate the evidence to the claims, not to the reference on which the Examiner relies. We have done exactly the mapping that the Examiner urges. (*Resp.* at 15-18) Our repeated reference in the Response to “removing McKendrick as a reference” was part of the conclusion that we drew for each claim; it did not relate to using McKendrick as a yardstick for what Glushko’s Slides, 1999 Article and 2005 Book needed to show. Therefore, the Examiner’s argument was misdirected. It also was mistaken.

The Board quoted a passage from *In re Stempel* (*On Request for Rehearing*, at 3) that has been widely interpreted as allowing Applicants to measure the sufficiency of their proof under rule 131 by reference to what the Examiner cited. The quoted passage from *In re Stempel* literally required us only to show “priority with respect to so much of the claimed invention as the [McKendrick] reference happens to show.” This implies that if the cited reference were thorough and detailed, applicants would need to submit thorough and detailed evidence. Conversely, when the Examiner relied on just 33 words in a popular press article and concluded that one of skill in the art would understand from just 33 words that Microsoft and others were using XML in a particular way, the Examiner set a VERY LOW hurdle for the Applicants to clear. *C.f., Ex Parte Jud, supra* (emphasizing study of application and references of record to determine level of skill in the art). If just 33 words convey that someone has reduced the claimed invention to practice, then the transcript of Glushko’s remarks, plus his slides, plus his technical

writings before any issue arose should overwhelmingly prove possession by Glushko's team of the invention before McKendrick. So it is not true, as the Examiner argued (FOA at 16), that we should ignore how the Examiner judged McKendrick. In colloquial terms, what's good for the goose is good for the gander. If 33 words from McKendrick arguably prove public possession of what we claim, our more extensive evidence is overwhelming.

From the evidence of Glushko's Slides, 1999 Article and 2005 Book, it is clear that, prior to July 25, 1998, Glushko's team was successfully running programs on computers that were within the scope of these claims. We mapped this RCE evidence to individual claims in our Response. Our evidence of record has not been criticized or diminished and therefore stands as a basis under *Ex Parte Foster* to remove McKendrick as a reference.

RCE Evidence Corroborates the Pre-Appeal Declarations

The Board's original decision (*Decision on Appeal*, at 8-9) criticized the pre-appeal declarations for not providing more evidence about the "first draft of CBL," which was mentioned in the corroborating status report. We responded to the Board raising this new issue during appeal by submitting historical texts that explained how these inventors spent 1998-99 educating those of skill in the art about "CBL". After the appeal was remitted, we submitted the evidence regarding CBL to the Examiner. In addition to Glushko's Slides, 1999 Article and the 2005 book, we presented Sall's Book, the Seybold Transcript, xCBL.org, Allen and Bosak as evidence regarding development of CBL in 1997 and read this corroborating evidence together with the pre-appeal declarations on the claims. (*Resp.* at 12-14 & 18-25)

The Examiner erred by failing to consider whether the RCE and appeal evidence corroborated the pre-appeal declarations. The Examiner's role was to study and judge the evidence that we presented. Judging is unlike making an argument, because all of the evidence needs to be discussed. In the final office action, the Examiner did not refer to, consider or weigh the evidence that we presented. That was reversible error.

The Examiner relied on non-declaration, non-prior art evidence as a basis for rejection, even while she ignored the non-declaration evidence that Applicants introduced. The Examiner cited the publication, Glushko, *How XML Enables Internet Training Communities and Marketplaces*", GCA Conferences, XML 99, Philadelphia

(Dec. 1999) accessible at <http://www.infoloom.com/gcaconfs/WEB/philadelphia99/glushko.HTM>. (FOA at 17) This publication 14 months after the application was filed is not prior art. The publication was selectively quoted, ignoring the immediately preceding paragraph that begins, “The oldest attempt to attack the problem of interoperability among vertical XML commerce applications is Commerce One's Common Business Library [CBL].” In this article, Glushko unequivocally attributed inventive work to Commerce One's predecessor, Veo and denied invention by others. The Examiner misevaluated this article, having failed to consider it as a whole.

The selected quote does not prove what the Examiner argues it does. It is better understood as saying that the master software architects at Veo developed a sophisticated architecture that worked, but decided to simplify it in a commercial product for so-called newbies to use. “Because of this research pedigree, early versions of CBL strove for logical completeness, expressiveness, and compactness to test the abstract modeling power of XML for electronic commerce and to identify requirements for development tools and runtime support. CBL 1.0 prototyping and application experience suggested that it was too abstract and powerful for XML ‘newbies’ and for people with traditional EDI backgrounds”. What this passage proves is that CBL 1.0, which predated CBL 1.1, ran in early versions (e.g., versions 0.7.2, 0.7.5 and Ingram /01), prototypes and applications. For instance, before September 1, 1998, CBL was used in “several demonstration projects, one with the federal government and one with the consortium of Japanese companies,” according to Dr. Glushko's remarks on September 1, 1998. (Ex. K; *Resp.* at 11) The demonstration projects were identified as GSA catalog interoperability and Project Seitai in Dr. Glushko's slide 31, on July 25, 1998. (Ex. I; *Resp.* at 11) The early versions of CBL provided experience for simplifying CBL in a commercial product, according to the quote. It appears that the Examiner applied the mistaken rule (FOA at 17 bottom), which would require Applicants to have completed a commercial product, rather than a prototype, in order to prove a reduction to practice. Again, as a matter of law, the prototype IS ENOUGH and a commercial product is NOT REQUIRED for an actual reduction to practice. *King Instrument, supra*, 767 F.2d at 861; *see, supra* at 15. The quote provides clear evidence of a tested prototype, which corroborates the pre-appeal declarations.

The evidence that we presented from Glushko's Slides, 1999 Article and 2005 book, and from Sall's Book, the Seybold Transcript, xCBL.org, Allen and Bosak (*Resp.* at 8-14) regarding Veo's ground breaking development in 1997 of CBL and related technologies was not criticized by the Examiner or minimized in any way. On this record, the evidence presented stands as credible and probative. It explains that working versions of CBL were available in late 1997 and early 1998. The excluded evidence completes the corroboration of the original declarations that the Board requested regarding "the first draft of CBL" and answers questions raised by the Board for the first time on appeal. In the RCE, we followed the Board's direction to submit our new evidence to the Examiner. The Examiner's decision to exclude evidence (*Resp.* at 8-14), which completes the corroboration of the pre-appeal declarations (*Resp.* at 18-25) and to ignore our reading of the corroborated pre-appeal declarations on the claims was reversible error.

RCE Evidence Establishes the 1998 Level of Skill in the Art as Immature, as Looking to These Inventors to Lead the Industry

The third set of evidence that we presented (*Resp.* at 25-34) related to the 1998 level of skill in the art, which determines the perspective from which the brief quotation that McKendrick attributed to Microsoft must be considered. We cited and explained *Ex Parte Jud, supra* (*Resp.* at 26-27), in which the Board explained how to judge the level of skill in the art.

The Examiner erred in not discussing the 1998 level of skill in the art in view of the new evidence. From *Graham v. John Deere* forward, discussing the level of skill in the art has been considered fundamental to analysis of obviousness. See, *KSR, supra*; *Ex Parte Jud, supra*. It again appears that the Examiner mistakenly relied on *res judicata*, because no other basis was given for excluding the new evidence. The Examiner's own opinion of how, with hindsight, the articles might be understood (FOA at 20) is not a legal substitute for reading McKendrick from the perspective of one of ordinary skill. Ignoring the 1998 level of skill in the art and expressing the Examiner's own opinion was reversible error.

One particularly telling bit of evidence from September 1, 1998 (McKendrick was published in September 1998) was a question from the audience to Dr. Glushko that asked whether XML was "stable enough" to warrant being used as a development

platform. (Ex. K; *Resp.* at 11) This evidence and the other evidence that we detailed (*Resp.* at 26-33) shows that these inventors were leading the industry in XML technology development, far ahead of the level of ordinary skill.

In the months immediately after W3C published the XML 1.0 recommendation, Microsoft described approaches that people might try in the future, not usually things that already had been done. (*Resp.* at 30-33) One of ordinary skill might have begun with report of Microsoft's vision from McKendrick or another source, but visionary puffery would quickly have given way to determining what tools Microsoft was providing and experimenting with the tools. (Exs. P-U) The logical path from Microsoft's musing forward would have been for one of ordinary skill to use RPC-XML tools that Microsoft was promoting and try to build with those tools. The technical documentation that we have provided shows that RPC-XML was a much different approach than we claimed.

The 1998 level of skill in the art combined with clear evidence that Microsoft was promoting RPC-XML, belies the conclusion of the Examiner and Board that the 33 words of McKendrick would have been understood by one of ordinary skill as enabling practice of the claimed data structure and method. Our 1998 level of skill evidence has not been criticized or diminished in any way. It proves that one of skill in the art would have understood Microsoft's comments about the future of XML, reported by McKendrick, as puffery or as promoting use of an RPC interface implemented with RPC-XML tools, and NOT as promoting or enabling a document style interface, which Veo was evangelizing at industry conferences. Therefore, the Examiner and Board's interpretation of McKendrick was mistaken because it was inadequately informed regarding the 1998 level of skill in the art. It was reversible error for the Examiner to ignore the 1998 level of skill evidence, to fail to apply perspective of one of ordinary skill, and to fail to reconsider the 33 words of McKendrick, given the new level of skill evidence.

CONCLUSION

Applicants respectfully submit that the pending claims are now in condition for allowance and thereby solicit acceptance of the claims as now stated.

Applicants strongly suggest an interview, because the record is extensive and packed with evidence. The undersigned can ordinarily be reached at his office at (650) 712-0340 from 8:30 a.m. to 5:30 p.m. PST, Monday through Friday, and can be reached at his cell phone at (415) 902-6112 most other times.

Fee Authorization. The Commissioner is hereby authorized to charge underpayment of any additional fees or credit any overpayment associated with this communication to Deposit Account No. 50-0869 (OIN 1004-1)

Respectfully submitted,

Dated: 21 July 2008

/Ernest J. Beffel, Jr./
Ernest J. Beffel, Jr.
Registration No. 43,489

Haynes Beffel & Wolfeld LLP
P.O. Box 366
Half Moon Bay, CA 94019
Telephone: (650) 712-0340
Facsimile: (650) 712-0263

**IV. Declarations Under Rule 131 Proving Reduction to Practice On or
Before January 21, 1998**

Decision on Petition Granted August 14, 2008

Inventors' Declarations

Declaration of Murray Maloney

Declaration of Bart Alan Meltzer

Declaration of John Glushko

Declaration of Matthew Fuchs

Non-Inventor's Declaration

Declaration of Kevin Hughes



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

Paper No.

HAYNES BEFFEL & WOLFELD LLP
P O BOX 366
HALF MOON BAY, CA 94019

COPY MAILED

AUG 14 2008

OFFICE OF PETITIONS

In re Application of :
Meltzer et al. : DECISION ON
Application No. 09/173,858 : PETITION
Filed: October 16, 1998 :
Atty Docket No. OIN 1004-1 :

This is a decision on the PETITION UNDER THE 37 CFR 1.183 REGARDING ENTRY OF RULE 131 DECLARATIONS BY FEWER THAN ALL INVENTORS filed July 21, 2008.

The petition under 37 CFR 1.183 is **GRANTED to the extent indicated herein.** This is not a decision on the merits of the 1.131 declaration.

The above-identified application was filed on October 16, 1998. A 37 CFR 1.63 declaration signed by all of the inventors (Meltzer, Allen, Fuchs, Glushko and Maloney) was filed on December 21, 1998. On July 21, 2008, applicants filed inventor's declarations under rule 131, by all of the inventors except inventor Allen.

Applicants have filed the instant petition to have the 37 CFR 1.131 declarations accepted without the signature of inventor Allen. Applicants request waiver of the signature of an unavailable inventor and acceptance of other evidence as well. Applicants state that inventor Allen signed the original 1.63 declaration, but refuses to sign the 1.131 declaration. Applicants state that inventor Allen previously refused to accept by mail a draft declaration and again is unavailable to sign an affidavit or declaration under 37 CFR 1.31, because he refused our mail. In support thereof, applicants submit a declaration of facts of Brianna Dahlberg, attesting to the

R-323

attempts to present the 1.131 declaration to inventor Allen, as well as, copies of the Express Mail envelopes which transmitted the 1.131 declaration as returned and refused by inventor Allen.

37 CFR 1.131 states, in pertinent part:

When any claim of an application or a patent under reexamination is rejected, the inventor of the subject matter of the rejected claim, the owner of the patent under reexamination, or the party qualified under §§ 1.42, 1.43, or 1.47, may submit an appropriate oath or declaration to establish invention of the subject matter of the rejected claim prior to the effective date of the reference or activity on which the rejection is based.

In addition, the Manual of Patent Examining Procedure states that "an application or declaration by less than all named inventors of an application is accepted where it is shown that less than all named inventors of an application invented the subject matter of the claim or claims under rejection."

Here, there has not been a party qualified under 37 CFR 1.42, 1.43, or 1.47. In addition, applicant does not contend that less than all of the named inventors of the application invented the subject matter of the claims under rejection. Accordingly, the proper parties to sign the 37 CFR 1.131 declaration include all of the joint inventors.

In order for a petition under 37 CFR 1.183 to be granted to waive the requirement that all of the joint inventors sign the 1.131 declaration, petitioner must demonstrate that this is an extraordinary situation where justice requires waiver of the rules.

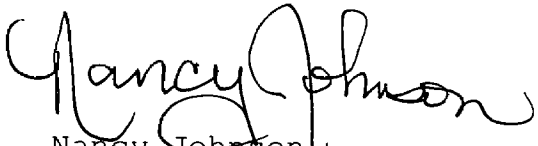
On instant petition, applicants have set forth the steps taken to obtain joint inventor Allen's signature on the 1.131 declaration. Applicants have shown that a bona fide effort was made to present the 1.131 declaration to inventor Allen and documentation supporting a conclusion that inventor Allen refused the presentation. Applicants have provided 1.131 declarations signed by all of the other joint inventors. Under the circumstances, it is concluded that petitioner has demonstrated that this is an extraordinary situation, warranting waiver of the rules.

The petition is granted to the extent that the 37 CFR 1.131 declaration may be entered, despite the fact that its requirement that all of the inventors sign the declaration has not been satisfied. This is not a decision on the merits of the declaration or on any other evidence presented.

The \$400 fee required for consideration of this petition under § 1.183 has been charged to Deposit Account No. 50-0869 (OIN 1004-1), as authorized.

Technology Center AU 2178 has been advised of this decision. The application is, thereby, forwarded to the Technology Center for further action by the examiner in light of this decision granting waiver of the requirement that all of the inventors sign the 1.131 declaration.

Telephone inquiries concerning this decision should be directed to the undersigned at (571) 272-3219.


Nancy Johnson
Senior Petitions Attorney
Office of Petitions

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Bart A. Meltzer

Application No.: 09/173,858

Confirmation No.: 4734

Filed: October 16, 1998

Art Unit: 2178

For: DOCUMENTS FOR COMMERCE IN
TRADING PARTNER NETWORKS AND
INTERFACE DEFINITIONS BASED ON THE
DOCUMENTS

Examiner: C. L. T. Huynh

**INVENTOR'S DECLARATION UNDER RULE 131 PROVING
REDUCTION TO PRACTICE ON OR BEFORE JANUARY 21, 1998**

I, Murray Maloney, declare as follows:

1. I am a former contractor to CN Group, Veo and Commerce One and a named inventor. All of the statements made herein are my own personal knowledge or of my own opinion, based on my training and experience, except where stated on information and belief. I could competently testify thereto if called as a witness.

2. This declaration is given in support of the application entitled "Documents for Commerce in Trading Partner Networks and Interface Definitions Based on the Documents," U.S. patent application serial number 09/173,858, filed on October 18, 1998, and any other applications in which it might be submitted. In the course of preparation of this declaration, I looked at the documents listed in the attached list of exhibits, a set of patent claims, and this patent application. I am informed and believe that some of the exhibits were obtained from archives maintained by my former colleague Kevin Hughes.

3. In beginning in 1997 and continuing through 1998, I contracted to do work for CN Group, which became Veo and merged into Commerce One. In this declaration, "CN Group" and "Veo" are used interchangeably without any effort to track down the date or circumstance of the name change. Commerce One is mentioned in passing, as the events discussed in this declaration preceded the date on which merger of Veo into Commerce One became effective. Years ago, I received stock and/or stock options in Commerce One, the majority of which I sold at a profit in 1999, 2000 and 2001.

4. I have been interested in the disposition of Commerce One's patents and patent applications, because they are very significant to business-to-business ecommerce. I am informed and believe that the Commerce One patents and patent applications were auctioned by a bankruptcy court and are now assigned to Open Invention Network ("OIN"), an organization formed by IBM, Phillips, Sony, Red Hat and Novell (and more recently joined by NEC), which has the web site www.openinventionnetwork.com. I am not involved in OIN, but I appreciate their stated dedication to acquire patents and offer them royalty free to promote Linux and spur innovation globally.

5. I have spoken with OIN's counsel Mr. Beffel more than one occasion but do not recall ever meeting him. I have not been compensated for this declaration.

6. Development of the technology disclosed in this patent application responded to a widely felt need for an object-oriented architectural framework for Internet commerce. The perceived need and efforts by IBM, Microsoft and others were described in an article by Tenenbaum, Jay M., Tripatinder S. Chowdhry and Kevin Hughes, "Eco System: An Internet Commerce Architecture" Computer May 1997: 48-55 ("Eco System article", submitted herewith, Exhibit A). The 1997 article proposed to use the conventional CORBA technology, which proved impractical.

7. By the time I started working with them in December, 2007, CN Group had already shifted from developing a CORBA architecture and to a document-based transaction interface architecture. That was fully two months before W3C published XML

as a recommendation in February 1998. CN Group's change in approach is explained in a second article by Glushko, Robert J., Jay M. Tenenbaum, Bart Meltzer, "An XML Framework for Agent-based E-commerce" Communications of the ACM, Vol. 42, No. 3, pp.106-109 & 111-114 (March 1999) ("XML Framework", submitted herewith, Exhibit B) that published after this application was filed. Note that several of the figures in the XML Framework article resemble figures used in the patent application.

8. Before this patent application was filed, we developed multiple versions of data structures that established how to define a document-based transaction interface architecture. We could see that these data structures, when placed in memory, would define an interface to a transaction using one or more input documents and one or more output documents. The interface defined by these data structures was practical and workable, in contrast to the CORBA architecture proposed in the 1997 article. We understood that these data structures also could drive a code generating process that would generate Java class definitions for internal data manipulation and translators between XML and Java for marshalling and unmarshalling data between external and internal formats. Unlike CORBA, the document-based transaction interface architecture involved programs communicating with one another in a human readable external format, while using a machine-friendly internal format. So-called marshalling and unmarshalling components translated between XML and Java.

9. This declaration presents two of the data structure versions developed to define a document-based transaction interface architecture, one for a demonstration and one presented during a conference. The earlier version showed a multi-stage transaction and specified several document exchanges as parts of the transaction. It is dated January 3, 1998, which is well in advance of a demonstration that was scheduled for January 21, 1998 (Exhibit F), which demonstration may have been postponed. The conference version was presented on July 25, 1998 (Exhibit I, slide 30).

10. The XML Framework article and this patent application explain how CBL was used to standardize interpretation data for XML documents. The resulting XML documents

are useful as input and output documents referenced in a data structure that defines a document-based transaction interface architecture. However, a document-based transaction interface need not be implemented using CBL and CBL can be used in other ways that do not involve using a document-based transaction interface.

11. Submitted with this declaration are files from three versions of CBL and infrastructure supporting transactions, which I am informed and believe were found in directories named 072, 075 and "ingram/01," and which bore file date stamps and internal documentary dates before January 21, 1998.

(a) An index.html file (Ex. C) from directory 072 lists many modules as included in version 0.7.2, before the end of 1997. The index.html file also provides a limited description of module testing and validation that had been completed before the end of 1997.

(b) I am informed and believe that reprinted files in Exhibit D come from the directory 072, even though some of them have earlier version numbers, such as catentry.dtd version 0.6 and cblcat.dtd version 0.6.1..

(c) The modules in directory 075 are internally documented as belonging to version 0.8, but I am informed and believe that another version of the modules appears in a directory 080, with later dates. I am informed and believe that reprinted files in Exhibit E come from directory 075.

(d) I am informed and believe that reprinted files in Exhibit H come from directory ingram/01.

(e) Before January 21, 1998, files and modules from the three versions of CBL and infrastructure supporting transactions represented in Exhibits C-E and H were available for use in a demonstration.

12. Exhibit D consists of files dated from November 2 through December 5, 1997 from the directory 072. It includes, in the file "command.dtd" dated December 5, 1997, definitions of an ENTITY and ELEMENTs used for registering documents and services in a registry. The ENTITY is named "command" and the ELEMENTs include "command.set", "register.document" and "register.service". Registering documents and services was part of the infrastructure that we developed for using a choreographed exchange of documents to conduct a transaction. That is, as of December 5, 1997, Exhibit D already includes the basic infrastructure defining the registration of documents and services in a registry.

13. Submitted as Exhibit F is a plan for preparing for a demonstration to Ingram Micro, entitled "Requirements and Tasks for the January Demo, (Updated 1/6/98 by Kenneth)". The plan called for "1-3 clients which run in a browser" and "1 server process". "All information exchange is done via CBL/XML streams ..." The demonstration scenario is described as follows, from pages 4 and 6 of the plan:

"1. Market Description. Ingram creates **imdesc.xml** and publicizes it. imdesc.xml refers to ingram.xml.

"2. IBM, DEC, COMPUSA, and Fry's send market participant info to Ingram so they can be approved and listed in the market's directories.

"ibm.xml, dec.xml, compu.xml, frys.xml

"Ingram acks and registers the documents.

"3. IBM and DEC send product descriptions to Ingram, think.xml and hinote.xml. Ingram acks and from those product descriptions produces its own catalogue entries for them. Those cat entries are TBS. (May need multiple files for the multiple configs of these products.)

"4. COMPUSA and Fry's inquire about laptops that Ingram has. NEED DETAILS. TBS: the information request docs and the responses.

"5. We pretend that both COMPUSA and Fry's buy some of each laptop.

"6. IBM and DEC inquire about laptops that Ingram has and has sold (= laptops in the channel?). TBS: the information request docs and the responses.

“(More details)

“1) Setting up the "master node" in the channel --defining Ingram's role and the services it will be providing (e.g registration, integrated catalogs, part number mapping, handling queries about prices and inventories, ordering)

“2) Registering companies to participate in the channel as either (a) manufacturers or (b) resellers [this means describing themselves (core metadata), their services (e.g., their catalog schemas), their forms (their business document schemas)]

“3) The reseller's view of the channel:

a) show me the integrated catalog

b) here's a part number: what is everyone else calling it

b) show me my price list

“4) The manufacturer's view of the channel:

a) show me (my items in) the channel

b) here's a part number: what is everyone else calling it

c) what prices are being charged for this part”

The plan for demonstration, on pages 4-5, lists many components, including “.mod” modules and “.dtd” data type definitions for XML documents. Components with names matching all of the names listed in the plan appear in the attached reprints from the 072, 075 and ingram/01 directories. Note that the scenario point # 1 is, “Ingram creates imdesc.xml and publicizes it.”

14. The file “imdesc.xml” (Exhibit G), which I am informed and believe comes from the “ingram/01” directory, is one version of a data structure that defines an interface to a transaction that includes multiple input documents and output documents. The file is written in XML and points to “.dtd” definitions of input and output documents for various services related to “Ordering and Fulfillment”. The file includes, in part, the text:

```
<!-- imdesc.xml Version: 0.1 -->
<!-- Purpose: marketplace description for Ingram Micro demo -->
<!-- Terry Allen 2 Jan 1998 -->
<!-- Copyright 1998 CNgroup, Inc. -->
<?xml version="1.0"?>
<!DOCTYPE market.description SYSTEM "imarkdsc.dtd">
<market.description>
```

```

<market.name>Ingram Micro Online Sales Network
</market.name>
<market.operator>
<market.operator.name>Ingram Micro Inc.
</market.operator.name>
<market.participant.info.pointer>
  <xll.locator urlink="ingram.xml">Ingram's Market Participant Info
  </xll.locator>
</market.participant.info.pointer>
</market.operator>
<service.set>
  <service>
    <service.name>Ordering and Fulfillment
    </service.name>
    <service.function.sequence>
      <service.function>
        <doctype from.party="any" to.party="ingram">order.dtd</doctype>
        <doctype from.party="ingram" to.party="any">ack.dtd</doctype>
      </service.function>
      <service.function>
        <doctype from.party="ingram" to.party="any">invoiceo.dtd</doctype>
        <doctype from.party="any" to.party="ingram">ack.dtd</doctype>
      </service.function>
      <service.function>
        <doctype from.party="ingram" to.party="any">shipnote.dtd</doctype>
      </service.function>
      <service.function>
        <doctype from.party="any" to.party="ingram">paynoteo.dtd</doctype>
        <doctype from.party="ingram" to.party="any">ack.dtd</doctype>
      </service.function>
    </service.function.sequence>
  </service>
</service.set> ...

```

The bold facing is added for emphasis. The internal documentation date of this file is January 2, 1998. I am informed and believe that the file date stamp indicates that the file was last changed on January 4, 1998 at 3:47 a.m. Both dates are well in advance of the scheduled demonstration on January 21, 1998.

15. The imdesc.xml file, in the excerpt above, describes service functions for "Ordering and Fulfillment" that are registered to be accessible through Ingram's URL. One

service receives an order and acknowledges it. Another generates an invoice and expects an acknowledgment. A third generates a shipping notice, without expecting an acknowledgment. Another receives a payment notice and acknowledges it. Each of these services is accessible through an interface that accepts an XML input document that conforms to a “.dtd” file and, typically, returns an XML acknowledgement document.

16. Before January 21, 1998, we had sufficiently worked with imdesc.xml to recognize and understand that it would serve its intended purpose of defining a document-based transaction interface that can include a series of related document exchanges. This file was one of many that were under development. The “index.html” (Ex. C) confirms that these files were tested during development. Exhibits D, E and H demonstrate the availability of the supporting files called for by the demonstration plan. By comparison of Exhibits D, E and H to the demonstration plan (Ex. F), one sees that all of the files listed as desired for the demonstration were available before January 21, 1998. Our intended use of the “imdesc.xml” file was as an interface definition. (Ex. F, quoted above) In January 1998, we correctly understood that the imdesc.xml data structure would work for its intended purpose. Later versions of our interface definition data structure, such as the version that appears in our patent application, generally followed the document interface definition pattern of “imdesc.xml”. The interface definition version (Exhibit I) that was publicly disclosed at a conference on July 25, 1998, at the International Workshop on Component-based Electronic Commerce was held at the Fisher Center for Management & Information Technology, Haas School of Business, UC Berkeley also followed the same pattern.

17. The July 25 version, Slide 30, Exhibit I included:

```
<service>
<service.name>Order Service</service.name>
  <service.location>www.veosystems.com/order</service.location>
  <service.op>
    <service.op.name>Submit Order</service.op.name>
    <service.op.inputdoc>po.dtd</service.op.inputdoc>
    <service.op.outputdoc>poack.dtd</service.op.outputdoc>
  </service.op>
```

```

    < service.op>
      < service.op.name>Track Order</service.op.name>
      <service.op.inputdoc>request.track.dtd<service.op.inputdoc>
      <service.op.outputdoc>response.track.dtd<service.op.outputdoc>
    </service.op>
  </service>

```

A very similar version appears in the patent application on page 45, which I am informed and believe also appears as CD-ROM appendix LISTING 5, after reformatting of specification.

18. Our January 1998 understanding that this data structure would work for its intended purpose was subsequently validated by others, who adopted our approach to interface definition. For instance, the later W3 "note" dated March 15, 2001 describing WSDL version 1.1 followed the same document-based interface pattern. In Exhibit J, taken from <http://www.w3.org/TR/wsdl>, section 1.2, WSDL Document Example, we see a simplified version of our interface definition, which involves a single document exchange, as opposed to a series of related document exchanges:

```

<portType name="StockQuotePortType">
  <operation name="GetLastTradePrice">
    <input message="tns:GetLastTradePriceInput"/>
    <output message="tns:GetLastTradePriceOutput"/>
  </operation>
</portType>

```

19. Next, we compare the two versions of our data structure to the pending claims. Singularly and collectively, the versions of data structures, which were stored in memory accessible to at least one node on a network, included the characteristics and features described below.

Claim 1: In Exhibits G-I, both imdesc.xml and Slide 30 depict machine readable interface specifications. The imdesc.xml file typically is found in a file directory on a

machine readable storage media. The PowerPoint Slide 30 appears to have been taken from a computer file similar to imdesc.xml and pasted into the presentation. The archive for imdesc.xml and the PowerPoint presentation both were stored on machine readable storage media. Both data structures define an interface to a transaction process. The imdesc.xml defines an interface with several service “functions”. Slide 30 defines multiple service “operations”. In imdesc.xml, one of the input documents is an “order”; in Slide 30, there is a “po”, which is short for purchase order. In both transaction interface definitions, an acknowledgement is sent, an “ack” in the earlier version and a “poack” in the later version. The input document definitions are referenced by “order.dtd”, “invoiceo.dtd”, “paynoteo.dtd”, “po.dtd” and “request.track.dtd”, which are data type definition (dtd) files. The output document definitions are referenced by “ack.dtd”, “poack.dtd” and “response.track.dtd”. The January 1998 demonstration scenario (Ex. F, quoted above) makes it clear that these interface definitions were published to nodes on a network that might desire to invoke the functions for which interfaces were defined. The context of the slides in Exhibit I make it clear that this interface was hosted and accessible to a plurality of nodes on a network in the development environment from which it was borrowed for the PowerPoint presentation.

Claim 2: The data type specifications in the exemplary “order.dtd” (Ex. E) include at least one logical structure. Similarly, slides 25-29 (Ex. I) depict using logical structure building blocks to construct documents such as a purchase order (po.dtd). In general, a data type definition file (dtd) will include data type specifications for at least on logical data structure.

Claim 3: The data type specifications for the country field of the exemplary “order.dtd” (Ex. E) include at least one data structure mapping predefined sets of storage units for a particular logical structure in the definitions to respective entries in a list. The country code field is supported by a list in “codes.mod” (the codes module), which is incorporated into addresso.mod which, in turn, is part of order.dtd. With a bit of tracing from imdesc.xml, one can see where the reusable lists were part of the system. Turning to

the July 25, 1998 presentation, slides 27-28 similarly illustrate use of country codes in a purchase order, defined by the file "po.dtd".

Claim 4: In Exhibit D, the ENTITY "command" and the related ELEMENTs support registering definitions of documents and services in a repository in memory accessible to at least one node. The definitions of documents and services include logical structures and interpretation information for logical structures. In Exhibit I, Slide 23 depicts a service registration document in a registry accessible to discovery nodes on a network. Slide 31 indicates that CBL was already in use for demonstration applications, Project Seitai and GSA catalog interoperability. We found it very useful to post our interface definitions for programmers to rely on during development, such as the imdesc.xml interface definition.

Claim 5: Looking through files from the ingram/01 directory, which are reproduced in Exhibit H, one finds the data type definition corresponding to the sample interface definition imdesc.xml. Beginning with imdesc.xml on page 32 of 66, one sees reference to "imarkdsc.dtd" in the DOCTYPE statement of line six. This data type definition appears on page 29 of 66, and incorporates by reference the "isrvprim.mod" module that appears on page 54 of 66. The isrvprim.mod defines elements of imdesc.xml, including service.name, service.location.pointer, and service.function. Thus, the sample "imdesc.xml" machine readable specification of Exhibit G complies with the imarkdsc.dtd definition of an interface document including logical structures for storing identifiers (e.g., service.name and/or input doctype) and references to definitions (dtd's) of input and output documents. Similarly, in Exhibit I, slide 30, identifiers of particular transactions appear as "Submit Order" and "Track Order." The references to definitions of the input and output documents are dtd file names. The "Loose Coupling" via Shared Document Definitions slide 22 makes it clear that the definition in slide 30 is a document that complies with a dtd, as in the ingram/01 files.

Claim 6: Both the "imdesc.xml" Exhibit G and the Slide 30 in Exhibit I specify one or more transactions supported by the interface.

Claim 7: Both the “imdesc.xml” Exhibit G and the Slide 30 in Exhibit I include references to documents used in the particular transactions, specifying the document dtd’s by name instead of providing a copy of them.

Claims 8-12: The input and output documents in both versions are depicted as XML documents. Generally, XML documents compliant with the February 1998 recommendation can be parsed or unparsed data. XML documents may encode text characters and the text characters may provide a natural language word. XML documents generally include markup data to identify sets of storage units.

Claim 13: The files reprinted in Exhibits D, E and H from directories 072, 075 and ingram/01 store document types that can be used in a plurality of transactions. Commands identified above were used to register these document types in a repository. Slides 25-29 depict a library of document types stored in a repository. The dtd definitions for the documents used in the transactions are referenced by names. For instance, in “imdesc.xml” we see the “ack.dtd” as the output document for several service functions.

Claim 14: The collections of files from directories include “imarkprt.dtd” and “imarkdsc.dtd” for market participant information and market description information. These modules are called out in Slide 28 of Exhibit I by slightly different and more easily readable names, “markpart.dtd” and “markdesc.dtd”. Some of these document types include identifications of participant processes.

Claim 15: The dtd files referenced in “imdesc.xml” and Slide 30, e.g., order.dtd and ack.dtd, are recognizable as compliant with a standard Extensible Markup Language XML.

Claim 16: The file “imdesc.xml” and the code excerpt in Slide 30 are recognizable as compliant with a standard Extensible Markup Language XML.

Claim 61: In the course of creating “imdesc.xml” and Slide 30, the authors of those documents went through the process of defining a machine readable interface

definition including an input and output document. The versions of the transaction interface definition data structure were provided to network nodes that requested the definition data structures.

Claim 62: The data type specifications in the exemplary “order.dtd” include at least one logical structure. Similarly, slides 25-29 depict using logical structure building blocks to construct documents such as a purchase order (po.dtd). In general, a data type definition file (dtd) will include data type specifications for at least on logical data structure.

Claim 63: The data type specifications for the country field of the exemplary “order.dtd” (Ex. E) include at least one data structure mapping predefined sets of storage units for a particular logical structure in the definitions to respective entries in a list. The country code field is supported by a list in “codes.mod” (the codes module), which is incorporated into addresso.mod which, in turn, is part of order.dtd. With a bit of tracing from imdesc.xml, one can see where the reusable lists were part of the system. Use of country codes in addresses is a feature of CBL that extended the XML recommendation – the XML recommendation uses codes for language identification and not for addresses. See, W3C Recommendation, Extensible Markup Language (XML) 1.0, § 2.12 (Feb. 10, 1998) accessed at <http://www.w3.org/TR/1998/REC-xml-19980210#dt-app>. Turning to the July 25, 1998 presentation, slides 27-28 similarly illustrate use of country codes in a purchase order, defined by the file “po.dtd”.

Claim 64: Slide 23 (Ex. I) depicts a service registration document in a registry accessible to nodes on a network. Slide 31 indicates that CBL was already in use for demonstration applications, Project Seitai and GSA catalog interoperability. We found it very useful to post our interface definitions for programmers to rely on during development, such as the imdesc.xml interface definition. In the January 1998 and earlier materials (Exs. D, E and H), we have printed some libraries of logical structures that we were using then. The Slides indicate that the libraries of logical structures were still in use six months later, when the conference presentation was made in July 1998.

Claim 65: Looking through files from the ingram/01 directory (Ex. H) one finds how imdesc.xml has been defined. Beginning with imdesc.xml on page 32 of 66, one sees reference to "imarkdsc.dtd" in the DOCTYPE statement of line six. This data type definition appears on page 29 of 66, and incorporates by reference the "isrvprim.mod" module that appears on page 54 of 66. The isrvprim.mod defines elements of imdesc.xml, including service.name, service.location.pointer, and service.function. Thus, the sample "imdesc.xml" machine readable specification of Exhibit G complies with the imarkdsc.dtd definition of an interface document including logical structures for storing identifiers (e.g., service.name and/or input doctype) and references to definitions (dtd's) of input and output documents. Similarly, in Exhibit I, slide 30, identifiers of particular transactions appear as "Submit Order" and "Track Order." The references to definitions of the input and output documents are dtd file names. The "Loose Coupling" via Shared Document Definitions slide 22 makes it clear that the definition in slide 30 is a document that complies with a dtd, as in the ingram/01 files.

Claims 66-70: The input and output documents in both versions (code and presentation) are depicted as XML documents. Generally, XML documents compliant with the February 1998 recommendation can be parsed or unparsed data. XML documents may encode text characters and the text characters may provide a natural language word. XML documents generally include markup data to identify sets of storage units.

Claim 71: The dtd files referenced in "imdesc.xml" and Slide 30 are recognizable as compliant with a standard Extensible Markup Language XML.

Claim 72: Our development work in 1997-98 included applying parsers to input documents, generating one or more Java event signals in response to the logical structure of the input documents, and having event listener programs respond to the event signals.

20. I am informed and believe that Examiner Huynh has questioned whether versions of CBL prior to the publicly released version 1.1 worked. In fact, CBL worked for

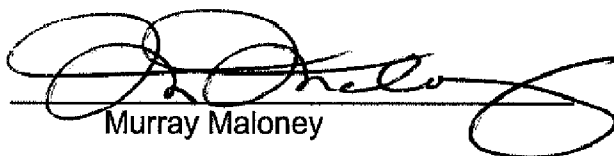
Application No.: 09/173,858

Docket No.: OIN 1004-1US

its intended purpose for many months and many versions prior to the September 1998 release of public version 1.1.

I declare under penalty of perjury of the laws of the United States of America that the foregoing is true and correct. I make this declaration with the understanding and knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 U.S.C. and that making willful false statements would jeopardize the validity of my application and any patents issuing thereon.

Executed this 17th day of JULY 2008 in GRAY HIGHLANDS, ONTARIO,
CANADA


Murray Maloney

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Bart A. Meltzer

Application No.: 09/173,858

Confirmation No.: 4734

Filed: October 16, 1998

Art Unit: 2178

For: DOCUMENTS FOR COMMERCE IN
TRADING PARTNER NETWORKS AND
INTERFACE DEFINITIONS BASED ON THE
DOCUMENTS

Examiner: C. L. T. Huynh

**INVENTOR'S DECLARATION UNDER RULE 131 PROVING
REDUCTION TO PRACTICE ON OR BEFORE JANUARY 21, 1998**

I, Bart Alan Meltzer, declare as follows:

1. I am a former employee of CN Group, Veo and Commerce One and a named inventor. All of the statements made herein are my own personal knowledge or of my own opinion, based on my training and experience, except where stated on information and belief. I could competently testify thereto if called as a witness.

2. This declaration is given in support of the application entitled "Documents for Commerce in Trading Partner Networks and Interface Definitions Based on the Documents," U.S. patent application serial number 09/173,858, filed on October 18, 1998, and any other applications in which it might be submitted. In the course of preparation of this declaration, I looked at the documents listed in the attached list of exhibits, a set of patent claims, and this patent application. I am informed and believe that some of the exhibits were obtained from archives maintained by my former colleague Kevin Hughes.

3. In beginning in 1997 and continuing through 1998, I worked for CN Group, which became Veo and merged into Commerce One. In this declaration, "CN Group" and "Veo" are used interchangeably without any effort to track down the date or circumstance of the name change. Commerce One is mentioned in passing, as the events discussed in this declaration preceded the date on which merger of Veo into Commerce One became effective. Years ago, I received stock and/or stock options in Commerce One, which I sold at a profit before Commerce One declared bankruptcy.

4. I have been interested in the disposition of Commerce One's patents and patent applications, because they are very significant to business-to-business ecommerce. I am informed and believe that the Commerce One patents and patent applications were auctioned by a bankruptcy court and are now assigned to Open Invention Network ("OIN"), an organization formed by IBM, Phillips, Sony, Red Hat and Novell (and more recently joined by NEC), which has the web site www.openinventionnetwork.com. I am not involved in OIN, but I appreciate their stated dedication to acquire patents and offer them royalty free to promote Linux and spur innovation globally.

5. I have met and spoken with OIN's counsel Mr. Beffel several times. I have not been compensated in any way for my efforts. When Commerce One was in bankruptcy, Mr. Beffel and I made an effort to buy Commerce One's portfolio, but we were entirely unsuccessful. The price paid for the patents was on the order of ten times as much as we were prepared to pay.

6. Development of the technology disclosed in this patent application responded to a widely felt need for an object-oriented architectural framework for Internet commerce. The perceived need and efforts by IBM, Microsoft and others were described in an article by Tenenbaum, Jay M., Tripatinder S. Chowdhry and Kevin Hughes, "Eco System: An Internet Commerce Architecture" Computer May 1997: 48-55 ("Eco System article", submitted herewith, Exhibit A). The 1997 article proposed to use the conventional CORBA technology, which proved impractical.

7. CN Group shifted from developing a CORBA architecture and to a document-based transaction interface architecture months before W3C published XML as a recommendation in February 1998. CN Group's change in approach is explained in a second article by Glushko, Robert J., Jay M. Tenenbaum, Bart Meltzer, "An XML Framework for Agent-based E-commerce" Communications of the ACM, Vol. 42, No. 3, pp.106-109 & 111-114 (March 1999) ("XML Framework", submitted herewith, Exhibit B) that published after this application was filed. Note that several of the figures in the XML Framework article resemble figures used in the patent application.

8. Before this patent application was filed, we developed multiple versions of data structures that established how to define a document-based transaction interface architecture. We could see that these data structures, when placed in memory, would define an interface to a transaction using one or more input documents and one or more output documents. The interface defined by these data structures was practical and workable, in contrast to the CORBA architecture proposed in the 1997 article. We understood that these data structures also could drive a code generating process that would generate Java class definitions for internal data manipulation and translators between XML and Java for marshalling and unmarshalling data between external and internal formats. Unlike CORBA, the document-based transaction interface architecture involved programs communicating with one another in a human readable external format, while using a machine-friendly internal format. So-called marshalling and unmarshalling components translated between XML and Java.

9. This declaration presents two of the data structure versions developed to define a document-based transaction interface architecture, one for a demonstration and one presented during a conference. The earlier version showed a multi-stage transaction and specified several document exchanges as parts of the transaction. It is dated January 3, 1998, which is well in advance of a demonstration that was scheduled for January 21, 1998 (Exhibit F), which demonstration may have been postponed. The conference version was presented on July 25, 1998 (Exhibit I, slide 30).

10. The XML Framework article and this patent application explain how CBL was used to standardize interpretation data for XML documents. The resulting XML documents are useful as input and output documents referenced in a data structure that defines a document-based transaction interface architecture. However, CBL need not be implemented using a document-based transaction interface.

11. Submitted with this declaration are files from three versions of CBL and infrastructure supporting transactions, which I am informed and believe were found in directories named 072, 075 and "ingram/01," and which bore file date stamps and internal documentary dates before January 21, 1998.

(a) An index.html file (Ex. C) from directory 072 lists many modules as included in version 0.7.2, before the end of 1997. The index.html file also provides a limited description of module testing and validation that had been completed before the end of 1997.

(b) I am informed and believe that reprinted files in Exhibit D come from the directory 072, even though some of them have earlier version numbers, such as catentry.dtd version 0.6 and cblcat.dtd version 0.6.1.

(c) The modules in directory 075 are internally documented as belonging to version 0.8, but I am informed and believe that another version of the modules appears in a directory 080, with later dates. I am informed and believe that reprinted files in Exhibit E come from directory 075.

(d) I am informed and believe that reprinted files in Exhibit H come from directory ingram/01.

(e) Before January 21, 1998, files and modules from the three versions of CBL and infrastructure supporting transactions represented in Exhibits C-E and H were available for use in a demonstration.

12. Exhibit D consists of files dated from November 2 through December 5, 1997 from the directory 072. It includes, in the file "command.dtd" dated December 5, 1997, definitions of an ENTITY and ELEMENTs used for registering documents and services in a registry. The ENTITY is named "command" and the ELEMENTs include "command.set", "register.document" and "register.service". Registering documents and services was part of the infrastructure that we developed for using a choreographed exchange of documents to conduct a transaction. That is, as of December 5, 1997, Exhibit D already includes the basic infrastructure defining the registration of documents and services in a registry.

13. Submitted as Exhibit F is a plan for preparing for a demonstration to Ingram Micro, entitled "Requirements and Tasks for the January Demo, (Updated 1/6/98 by Kenneth)". The plan called for "1-3 clients which run in a browser" and "1 server process". "All information exchange is done via CBL/XML streams ..." The demonstration scenario is described as follows, from pages 4 and 6 of the plan:

"1. Market Description. Ingram creates **imdesc.xml** and publicizes it. imdesc.xml refers to ingram.xml.

"2. IBM, DEC, COMPUSA, and Fry's send market participant info to Ingram so they can be approved and listed in the market's directories.

"ibm.xml, dec.xml, compu.xml, frys.xml

"Ingram acks and registers the documents.

"3. IBM and DEC send product descriptions to Ingram, think.xml and hinote.xml. Ingram acks and from those product descriptions produces its own catalogue entries for them. Those cat entries are TBS. (May need multiple files for the multiple configs of these products.)

"4. COMPUSA and Fry's inquire about laptops that Ingram has. NEED DETAILS. TBS: the information request docs and the responses.

"5. We pretend that both COMPUSA and Fry's buy some of each laptop.

"6. IBM and DEC inquire about laptops that Ingram has and has sold (= laptops in the channel?). TBS: the information request docs and the responses.

“(More details)

“1) Setting up the "master node" in the channel --defining Ingram's role and the services it will be providing (e.g registration, integrated catalogs, part number mapping, handling queries about prices and inventories, ordering)

“2) Registering companies to participate in the channel as either (a) manufacturers or (b) resellers [this means describing themselves (core metadata), their services (e.g., their catalog schemas), their forms (their business document schemas)]

“3) The reseller's view of the channel:

a) show me the integrated catalog

b) here's a part number: what is everyone else calling it

b) show me my price list

“4) The manufacturer's view of the channel:

a) show me (my items in) the channel

b) here's a part number: what is everyone else calling it

c) what prices are being charged for this part”

The plan for demonstration, on pages 4-5, lists many components, including “.mod” modules and “.dtd” data type definitions for XML documents. Components with names matching all of the names listed in the plan appear in the attached reprints from the 072, 075 and ingram/01 directories. Note that the scenario point # 1 is, “Ingram creates imdesc.xml and publicizes it.”

14. The file “imdesc.xml” (Exhibit G), which I am informed and believe comes from the “ingram/01” directory, is one version of a data structure that defines an interface to a transaction that includes multiple input documents and output documents. The file is written in XML and points to “.dtd” definitions of input and output documents for various services related to “Ordering and Fulfillment”. The file includes, in part, the text:

```
<!-- imdesc.xml Version: 0.1 -->
<!-- Purpose: marketplace description for Ingram Micro demo -->
<!-- Terry Allen 2 Jan 1998 -->
<!-- Copyright 1998 CNgroup, Inc. -->
<?xml version="1.0"?>
<!DOCTYPE market.description SYSTEM "imarkdsc.dtd">
<market.description>
```

```

<market.name>Ingram Micro Online Sales Network
</market.name>
<market.operator>
<market.operator.name>Ingram Micro Inc.
</market.operator.name>
<market.participant.info.pointer>
  <xll.locator urlink="ingram.xml">Ingram's Market Participant Info
  </xll.locator>
</market.participant.info.pointer>
</market.operator>
<service.set>
  <service>
    <service.name>Ordering and Fulfillment
    </service.name>
    <service.function.sequence>
      <service.function>
        <doctype from.party="any" to.party="ingram">order.dtd</doctype>
        <doctype from.party="ingram" to.party="any">ack.dtd</doctype>
      </service.function>
      <service.function>
        <doctype from.party="ingram" to.party="any">invoiceo.dtd</doctype>
        <doctype from.party="any" to.party="ingram">ack.dtd</doctype>
      </service.function>
      <service.function>
        <doctype from.party="ingram" to.party="any">shipnote.dtd</doctype>
      </service.function>
      <service.function>
        <doctype from.party="any" to.party="ingram">paynoteo.dtd</doctype>
        <doctype from.party="ingram" to.party="any">ack.dtd</doctype>
      </service.function>
    </service.function.sequence>
  </service>
</service.set> ...

```

The bold facing is added for emphasis. The internal documentation date of this file is January 2, 1998. I am informed and believe that the file date stamp indicates that the file was last opened on January 4, 1998 at 3:47 a.m. Both dates are well in advance of the scheduled demonstration on January 21, 1998.

15. The imdesc.xml file, in the excerpt above, describes service functions for "Ordering and Fulfillment" that are registered to be accessible through Ingram's URL. One

service receives an order and acknowledges it. Another generates an invoice and expects an acknowledgment. A third generates a shipping notice, without expecting an acknowledgment. Another receives a payment notice and acknowledges it. Each of these services is accessible through an interface that accepts an XML input document that conforms to a “.dtd” file and, typically, returns an XML acknowledgement document.

16. Before January 21, 1998, we had sufficiently worked with imdesc.xml to recognize and understand that it would serve its intended purpose of defining a document-based transaction interface that can include a series of related document exchanges. This file was one of many that were under development. The “index.html” (Ex. C) confirms that these files were tested during development. Exhibits D, E and H demonstrate the availability of the supporting files called for by the demonstration plan. By comparison of Exhibits D, E and H to the demonstration plan (Ex. F), one sees that all of the files listed as desired for the demonstration were available before January 21, 1998. Our intended use of the “imdesc.xml” file was as an interface definition. (Ex. F, quoted above) In January 1998, we correctly understood that the imdesc.xml data structure would work for its intended purpose. Later versions of our interface definition data structure, such as the version that appears in our patent application, generally followed the document interface definition pattern of “imdesc.xml”. The interface definition version (Exhibit I) that was publicly disclosed at a conference on July 25, 1998, at the International Workshop on Component-based Electronic Commerce was held at the Fisher Center for Management & Information Technology, Haas School of Business, UC Berkeley also followed the same pattern.

17. The July 25 version, Slide 30, Exhibit I included:

```
<service>
<service.name>Order Service</service.name>
  <service.location>www.veosystems.com/order</service.location>
  <service.op>
    <service.op.name>Submit Order</service.op.name>
    <service.op.inputdoc>po.dtd</service.op.inputdoc>
    <service.op.outputdoc>poack.dtd</service.op.outputdoc>
  </service.op>
```

```

    < service.op>
      < service.op.name>Track Order</service.op.name>
      <service.op.inputdoc>request.track.dtd<service.op.inputdoc>
      <service.op.outputdoc>response.track.dtd<service.op.outputdoc>
    </service.op>
  </service>

```

A very similar version appears in the patent application on page 45, which I am informed and believe also appears as CD-ROM appendix LISTING 5, after reformatting of specification.

18. Our January 1998 understanding that this data structure would work for its intended purpose was subsequently validated by others, who adopted our approach to interface definition. For instance, the later W3 "note" dated March 15, 2001 describing WSDL version 1.1 followed the same document-based interface pattern. In Exhibit J, taken from <http://www.w3.org/TR/wsdl>, section 1.2, WSDL Document Example, we see a simplified version of our interface definition, which involves a single document exchange, as opposed to a series of related document exchanges:

```

<portType name="StockQuotePortType">
  <operation name="GetLastTradePrice">
    <input message="tns:GetLastTradePriceInput"/>
    <output message="tns:GetLastTradePriceOutput"/>
  </operation>
</portType>

```

19. Next, we compare the two versions of our data structure to the pending claims. Singularly and collectively, the versions of data structures, which were stored in memory accessible to at least one node on a network, included the characteristics and features described below.

Claim 1: In Exhibits G-I, both imdesc.xml and Slide 30 depict machine readable interface specifications. The imdesc.xml file typically is found in a file directory on a

machine readable storage media. The Powerpoint Slide 30 appears to have been taken from a computer file similar to imdesc.xml and pasted into the presentation. The archive for imdesc.xml and the Powerpoint presentation both were stored on machine readable storage media. Both data structures define an interface to a transaction process. The imdesc.xml defines an interface with several service “functions”. Slide 30 defines multiple service “operations”. In imdesc.xml, one of the input documents is an “order”; in Slide 30, there is a “po”, which is short for purchase order. In both transaction interface definitions, an acknowledgement is sent, an “ack” in the earlier version and a “poack” in the later version. The input document definitions are referenced by “order.dtd”, “invoiceo.dtd”, “paynoteo.dtd”, “po.dtd” and “request.track.dtd”, which are data type definition (dtd) files. The output document definitions are referenced by “ack.dtd”, “poack.dtd” and “response.track.dtd”. The January 1998 demonstration scenario (Ex. F, quoted above) makes it clear that these interface definitions were published to nodes on a network that might desire to invoke the functions for which interfaces were defined. The context of the slides and other remarks by Glushko, including ongoing demonstration projects, make it clear that this interface was hosted and accessible to a plurality of nodes on a network in the development environment from which it was borrowed for the Powerpoint presentation.

Claim 2: The data type specifications in the exemplary “order.dtd” (Ex. E) include at least one logical structure. Similarly, slides 25-29 (Ex. I) depict using logical structure building blocks to construct documents such as a purchase order (po.dtd). In general, a data type definition file (dtd) will include data type specifications for at least one logical data structure.

Claim 3: The data type specifications for the country field of the exemplary “order.dtd” (Ex. E) include at least one data structure mapping predefined sets of storage units for a particular logical structure in the definitions to respective entries in a list. The country code field is supported by a list in “codes.mod” (the codes module), which is incorporated into addresso.mod which, in turn, is part of order.dtd. With a bit of tracing from imdesc.xml, one can see where the reusable lists were part of the system. Use of

country codes in addresses is a feature of CBL that extended the XML recommendation – the XML recommendation uses codes for language identification and not for addresses. See, W3C Recommendation, Extensible Markup Language (XML) 1.0, § 2.12 (Feb. 10, 1998) accessed at <http://www.w3.org/TR/1998/REC-xml-19980210#dt-app>. Turning to the July 25, 1998 presentation, slides 27-28 similarly illustrate use of country codes in a purchase order, defined by the file “po.dtd”.

Claim 4: In Exhibit D, the ENTITY “command” and the related ELEMENTs support registering definitions of documents and services in a repository in memory accessible to at least one node. The definitions of documents and services include logical structures and interpretation information for logical structures. In Exhibit I, Slide 23 depicts a service registration document in a registry accessible to discovery nodes on a network. Slide 31 indicates that CBL was already in use for demonstration applications, Project Seitai and GSA catalog interoperability. We found it very useful to post our interface definitions for programmers to rely on during development, such as the imdesc.xml interface definition.

Claim 5: Looking through files from the ingram/01 directory, which are reproduced in Exhibit H, one finds the data type definition corresponding to the sample interface definition imdesc.xml. Beginning with imdesc.xml on page 32 of 66, one sees reference to “imarkdsc.dtd” in the DOCTYPE statement of line six. This data type definition appears on page 29 of 66, and incorporates by reference the “isrvprim.mod” module that appears on page 54 of 66. The isrvprim.mod defines elements of imdesc.xml, including service.name, service.location.pointer, and service.function. Thus, the sample “imdesc.xml” machine readable specification of Exhibit G complies with the imarkdsc.dtd definition of an interface document including logical structures for storing identifiers (e.g., service.name and/or input doctype) and references to definitions (dtd’s) of input and output documents. Similarly, in Exhibit I, slide 30, identifiers of particular transactions appear as “Submit Order” and “Track Order.” The references to definitions of the input and output documents are dtd file names. The “Loose Coupling” via Shared Document Definitions

slide 22 makes it clear that the definition in slide 30 is a document that complies with a dtd, as in the ingram/01 files.

Claim 6: Both the “imdesc.xml” Exhibit G and the Slide 30 in Exhibit I specify one or more transactions supported by the interface.

Claim 7: Both the “imdesc.xml” Exhibit G and the Slide 30 in Exhibit I include references to documents used in the particular transactions, specifying the document dtd’s by name instead of providing a copy of them.

Claims 8-12: The input and output documents in both versions are depicted as XML documents. Generally, XML documents compliant with the February 1998 recommendation can be parsed or unparsed data. XML documents may encode text characters and the text characters may provide a natural language word. XML documents generally include markup data to identify sets of storage units.

Claim 13: The files reprinted in Exhibits D, E and H from directories 072, 075 and ingram/01 store document types that can be used in a plurality of transactions. Commands identified above were used to register these document types in a repository. Slides 25-29 depict a library of document types stored in a repository. The dtd definitions for the documents used in the transactions are referenced by names. For instance, in “imdesc.xml” we see the “ack.dtd” as the output document for several service functions.

Claim 14: The collections of files from directories include “imarkprt.dtd” and “imarkdsc.dtd” for market participant information and market description information. These modules are called out in Slide 28 of Exhibit I by slightly different and more easily readable names, “markpart.dtd” and “markdesc.dtd”. Some of these document types include identifications of participant processes.

Claim 15: The dtd files referenced in “imdesc.xml” and Slide 30, e.g., order.dtd and ack.dtd, are recognizable as compliant with a standard Extensible Markup Language XML.

Claim 16: The file “imdesc.xml” and the code excerpt in Slide 30 are recognizable as compliant with a standard Extensible Markup Language XML.

Claim 61: In the course of creating “imdesc.xml” and Slide 30, the authors of those documents went through the process of defining a machine readable interface definition including an input and output document. The versions of the transaction interface definition data structure were provided to network nodes that requested the definition data structures.

Claim 62: The data type specifications in the exemplary “order.dtd” include at least one logical structure. Similarly, slides 25-29 depict using logical structure building blocks to construct documents such as a purchase order (po.dtd). In general, a data type definition file (dtd) will include data type specifications for at least one logical data structure.

Claim 63: The data type specifications for the country field of the exemplary “order.dtd” (Ex. E) include at least one data structure mapping predefined sets of storage units for a particular logical structure in the definitions to respective entries in a list. The country code field is supported by a list in “codes.mod” (the codes module), which is incorporated into addresso.mod which, in turn, is part of order.dtd. With a bit of tracing from imdesc.xml, one can see where the reusable lists were part of the system. Use of country codes in addresses is a feature of CBL that extended the XML recommendation – the XML recommendation uses codes for language identification and not for addresses. See, W3C Recommendation, Extensible Markup Language (XML) 1.0, § 2.12 (Feb. 10, 1998) accessed at <http://www.w3.org/TR/1998/REC-xml-19980210#dt-app>. Turning to the July 25, 1998 presentation, slides 27-28 similarly illustrate use of country codes in a purchase order, defined by the file “po.dtd”.

Claim 64: Slide 23 (Ex. I) depicts a service registration document in a registry accessible to discovery nodes on a network. Slide 31 indicates that CBL was already in use for demonstration applications, Project Seitai and GSA catalog interoperability. We found it very useful to post our interface definitions for programmers to rely on during

development, such as the imdesc.xml interface definition. In the January 1998 and earlier materials (Exs. D, E and H), we have printed some libraries of logical structures that we were using then. The Slides indicate that the libraries of logical structures were still in use six months later, when the conference presentation was made in July 1998.

Claim 65: Looking through files from the ingram/01 directory (Ex. H) one finds how imdesc.xml has been defined. Beginning with imdesc.xml on page 32 of 66, one sees reference to "imarkdsc.dtd" in the DOCTYPE statement of line six. This data type definition appears on page 29 of 66, and incorporates by reference the "isrvprim.mod" module that appears on page 54 of 66. The isrvprim.mod defines elements of imdesc.xml, including service.name, service.location.pointer, and service.function. Thus, the sample "imdesc.xml" machine readable specification of Exhibit G complies with the imarkdsc.dtd definition of an interface document including logical structures for storing identifiers (e.g., service.name and/or input doctype) and references to definitions (dtd's) of input and output documents. Similarly, in Exhibit I, slide 30, identifiers of particular transactions appear as "Submit Order" and "Track Order." The references to definitions of the input and output documents are dtd file names. The "Loose Coupling" via Shared Document Definitions slide 22 makes it clear that the definition in slide 30 is a document that complies with a dtd, as in the ingram/01 files.

Claims 66-70: The input and output documents in both versions (code and presentation) are depicted as XML documents. Generally, XML documents compliant with the February 1998 recommendation can be parsed or unparsed data. XML documents may encode text characters and the text characters may provide a natural language word. XML documents generally include markup data to identify sets of storage units.

Claim 71: The dtd files referenced in "imdesc.xml" and Slide 30 are recognizable as compliant with a standard Extensible Markup Language XML.

Claim 72: Our development work in 1997-98 included applying parsers to input documents, generating one or more Java event signals in response to the logical structure of the input documents, and having event listener programs respond to the event signals.

20. I am informed and believe that Examiner Huynh has questioned whether versions of CBL prior to the publicly released version 1.1 worked. In fact, CBL worked for its intended purpose for many months and many versions prior to the September 1998 release of public version 1.1.

I declare under penalty of perjury of the laws of the United States of America that the foregoing is true and correct. I make this declaration with the understanding and knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 U.S.C. and that making willful false statements would jeopardize the validity of my application and any patents issuing thereon.

Executed this 7th day of July, 2008 in Corvallis, Ca.


Bart Alan Meltzer

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Bart A. Meltzer

Application No.: 09/173,858

Confirmation No.: 4734

Filed: October 16, 1998

Art Unit: 2178

For: DOCUMENTS FOR COMMERCE IN
TRADING PARTNER NETWORKS AND
INTERFACE DEFINITIONS BASED ON THE
DOCUMENTS

Examiner: C. L. T. Huynh

**INVENTOR'S DECLARATION UNDER RULE 131 PROVING
REDUCTION TO PRACTICE ON OR BEFORE JANUARY 21, 1998**

I, Robert John Glushko, declare as follows:

1. I am a former employee of CN Group, Veo and Commerce One and a named inventor. I currently am an Adjunct Full Professor at the University of California at Berkeley in the School of Information, the Director of the Center for Document Engineering, and one of the founding faculty members of the Information & Service Design program. With Tim McGrath, I am a co-author of the book Document Engineering (MIT Press, 2005). All of the statements made herein are my own personal knowledge or of my own opinion, based on my training and experience, except where stated on information and belief. I could competently testify thereto if called as a witness.

2. This declaration is given in support of the application entitled "Documents for Commerce in Trading Partner Networks and Interface Definitions Based on the Documents," U.S. patent application serial number 09/173,858, filed on October 18, 1998, and any other applications in which it might be submitted. In the course of preparation of this declaration, I looked at the documents listed in the attached list of exhibits, a set of

patent claims, and this patent application. I am informed and believe that some of the exhibits were obtained from archives maintained by my former colleague Kevin Hughes.

3. In beginning in 1997 and continuing through 1998, I worked for CN Group, which became Veo and merged into Commerce One. In this declaration, "CN Group" and "Veo" are used interchangeably without any effort to track down the date or circumstance of the name change. Commerce One is mentioned in passing, as the events discussed in this declaration preceded the date on which merger of Veo into Commerce One became effective. Years ago, I received stock and/or stock options in Commerce One, which I sold at a profit before Commerce One declared bankruptcy.

4. I have been interested in the disposition of Commerce One's patents and patent applications, because they are very significant to business-to-business ecommerce. I am informed and believe that the Commerce One patents and patent applications were auctioned by a bankruptcy court and are now assigned to Open Invention Network ("OIN"), an organization formed by IBM, Phillips, Sony, Red Hat and Novell (and more recently joined by NEC), which has the web site www.openinventionnetwork.com. I am not involved in OIN, but I appreciate their stated dedication to acquire patents and offer them royalty free to promote Linux and spur innovation globally.

5. I have met and spoken with OIN's counsel Mr. Beffel more than one occasion. OIN paid for my lunch once, when I met with Mr. Beffel and OIN's Jerry Rosenthal. I have not been compensated in any way for my efforts.

6. Development of the technology disclosed in this patent application responded to a widely felt need for an object-oriented architectural framework for Internet commerce. The perceived need and efforts by IBM, Microsoft and others were described in an article by Tenenbaum, Jay M., Tripatinder S. Chowdhry and Kevin Hughes, "Eco System: An Internet Commerce Architecture" Computer May 1997: 48-55 ("Eco System article", submitted herewith, Exhibit A). The 1997 article proposed to use the conventional CORBA technology, which proved impractical.

7. CN Group shifted from developing a CORBA architecture and to a document-based transaction interface architecture months before W3C published XML as a recommendation in February 1998. CN Group's change in approach is explained in a second article by Glushko, Robert J., Jay M. Tenenbaum, Bart Meltzer, "An XML Framework for Agent-based E-commerce" Communications of the ACM, Vol. 42, No. 3, pp.106-109 & 111-114 (March 1999) ("XML Framework", submitted herewith, Exhibit B) that published after this application was filed. Note that several of the figures in the XML Framework article resemble figures used in the patent application.

8. Before this patent application was filed, we developed multiple versions of data structures that established how to define a document-based transaction interface architecture. We could see that these data structures, when placed in memory, would define an interface to a transaction using one or more input documents and one or more output documents. The interface defined by these data structures was practical and workable, in contrast to the CORBA architecture proposed in the 1997 article. We understood that these data structures also could drive a code generating process that would generate Java class definitions for internal data manipulation and translators between XML and Java for marshalling and unmarshalling data between external and internal formats. Unlike CORBA, the document-based transaction interface architecture involved programs communicating with one another in a human readable external format, while using a machine-friendly internal format. So-called marshalling and unmarshalling components translated between XML and Java.

9. This declaration presents two of the data structure versions developed to define a document-based transaction interface architecture, one for a demonstration and one presented during a conference. The earlier version showed a multi-stage transaction and specified several document exchanges as parts of the transaction. It is dated January 3, 1998, which is well in advance of a demonstration that was scheduled for January 21, 1998 (Exhibit F), which demonstration may have been postponed. The conference version was presented on July 25, 1998 (Exhibit I, slide 30).

10. The XML Framework article and this patent application explain how CBL was used to standardize interpretation data for XML documents. The resulting XML documents are useful as input and output documents referenced in a data structure that defines a document-based transaction interface architecture. However, a document-based transaction interface need not be implemented using CBL and CBL can be used in other ways that do not involve using a document-based transaction interface.

11. Submitted with this declaration are files from three versions of CBL and infrastructure supporting transactions, which I am informed and believe were found in directories named 072, 075 and "ingram/01," and which bore file date stamps and internal documentary dates before January 21, 1998.

(a) An index.html file (Ex. C) from directory 072 lists many modules as included in version 0.7.2, before the end of 1997. The index.html file also provides a limited description of module testing and validation that had been completed before the end of 1997.

(b) I am informed and believe that reprinted files in Exhibit D come from the directory 072, even though some of them have earlier version numbers, such as catentry.dtd version 0.6 and cblcat.dtd version 0.6.1..

(c) The modules in directory 075 are internally documented as belonging to version 0.8, but I am informed and believe that another version of the modules appears in a directory 080, with later dates. I am informed and believe that reprinted files in Exhibit E come from directory 075.

(d) I am informed and believe that reprinted files in Exhibit H come from directory ingram/01.

(e) Before January 21, 1998, files and modules from the three versions of CBL and infrastructure supporting transactions represented in Exhibits C-E and H were available for use in a demonstration.

12. Exhibit D consists of files dated from November 2 through December 5, 1997 from the directory 072. It includes, in the file "command.dtd" dated December 5, 1997, definitions of an ENTITY and ELEMENTs used for registering documents and services in a registry. The ENTITY is named "command" and the ELEMENTs include "command.set", "register.document" and "register.service". Registering documents and services was part of the infrastructure that we developed for using a choreographed exchange of documents to conduct a transaction. That is, as of December 5, 1997, Exhibit D already includes the basic infrastructure defining the registration of documents and services in a registry.

13. Submitted as Exhibit F is a plan for preparing for a demonstration to Ingram Micro, entitled "Requirements and Tasks for the January Demo, (Updated 1/6/98 by Kenneth)". The plan called for "1-3 clients which run in a browser" and "1 server process". "All information exchange is done via CBL/XML streams ..." The demonstration scenario is described as follows, from pages 4 and 6 of the plan:

"1. Market Description. Ingram creates **imdesc.xml** and publicizes it. imdesc.xml refers to ingram.xml.

"2. IBM, DEC, COMPUSA, and Fry's send market participant info to Ingram so they can be approved and listed in the market's directories.

"ibm.xml, dec.xml, compu.xml, frys.xml

"Ingram acks and registers the documents.

"3. IBM and DEC send product descriptions to Ingram, think.xml and hinote.xml. Ingram acks and from those product descriptions produces its own catalogue entries for them. Those cat entries are TBS. (May need multiple files for the multiple configs of these products.)

"4. COMPUSA and Fry's inquire about laptops that Ingram has. NEED DETAILS. TBS: the information request docs and the responses.

"5. We pretend that both COMPUSA and Fry's buy some of each laptop.

"6. IBM and DEC inquire about laptops that Ingram has and has sold (= laptops in the channel?). TBS: the information request docs and the responses.

“(More details)

“1) Setting up the "master node" in the channel --defining Ingram's role and the services it will be providing (e.g registration, integrated catalogs, part number mapping, handling queries about prices and inventories, ordering)

“2) Registering companies to participate in the channel as either (a) manufacturers or (b) resellers [this means describing themselves (core metadata), their services (e.g., their catalog schemas), their forms (their business document schemas)]

“3) The reseller's view of the channel:

a) show me the integrated catalog

b) here's a part number: what is everyone else calling it

b) show me my price list

“4) The manufacturer's view of the channel:

a) show me (my items in) the channel

b) here's a part number: what is everyone else calling it

c) what prices are being charged for this part”

The plan for demonstration, on pages 4-5, lists many components, including “.mod” modules and “.dtd” data type definitions for XML documents. Components with names matching all of the names listed in the plan appear in the attached reprints from the 072, 075 and ingram/01 directories. Note that the scenario point # 1 is, “Ingram creates imdesc.xml and publicizes it.”

14. The file “imdesc.xml” (Exhibit G), which I am informed and believe comes from the “ingram/01” directory, is one version of a data structure that defines an interface to a transaction that includes multiple input documents and output documents. The file is written in XML and points to “.dtd” definitions of input and output documents for various services related to “Ordering and Fulfillment”. The file includes, in part, the text:

```
<!-- imdesc.xml Version: 0.1 -->
<!-- Purpose: marketplace description for Ingram Micro demo -->
<!-- Terry Allen 2 Jan 1998 -->
<!-- Copyright 1998 CNgroup, Inc. -->
<?xml version="1.0"?>
<!DOCTYPE market.description SYSTEM "imarkdsc.dtd">
<market.description>
```

```

<market.name>Ingram Micro Online Sales Network
</market.name>
<market.operator>
<market.operator.name>Ingram Micro Inc.
</market.operator.name>
<market.participant.info.pointer>
  <xll.locator urlink="ingram.xml">Ingram's Market Participant Info
  </xll.locator>
</market.participant.info.pointer>
</market.operator>
<service.set>
  <service>
    <service.name>Ordering and Fulfillment
    </service.name>
    <service.function.sequence>
      <service.function>
        <doctype from.party="any" to.party="ingram">order.dtd</doctype>
        <doctype from.party="ingram" to.party="any">ack.dtd</doctype>
      </service.function>
      <service.function>
        <doctype from.party="ingram" to.party="any">invoiceo.dtd</doctype>
        <doctype from.party="any" to.party="ingram">ack.dtd</doctype>
      </service.function>
      <service.function>
        <doctype from.party="ingram" to.party="any">shipnote.dtd</doctype>
      </service.function>
      <service.function>
        <doctype from.party="any" to.party="ingram">paynoteo.dtd</doctype>
        <doctype from.party="ingram" to.party="any">ack.dtd</doctype>
      </service.function>
    </service.function.sequence>
  </service>
</service.set> ...

```

The bold facing is added for emphasis. The internal documentation date of this file is January 2, 1998. I am informed and believe that the file date stamp indicates that the file was last changed on January 4, 1998 at 3:47 a.m. Both dates are well in advance of the scheduled demonstration on January 21, 1998.

15. The imdesc.xml file, in the excerpt above, describes service functions for "Ordering and Fulfillment" that are registered to be accessible through Ingram's URL. One

service receives an order and acknowledges it. Another generates an invoice and expects an acknowledgment. A third generates a shipping notice, without expecting an acknowledgement. Another receives a payment notice and acknowledges it. Each of these services is accessible through an interface that accepts an XML input document that conforms to a “.dtd” file and, typically, returns an XML acknowledgement document.

16. Before January 21, 1998, we had sufficiently worked with imdesc.xml to recognize and understand that it would serve its intended purpose of defining a document-based transaction interface that can include a series of related document exchanges. This file was one of many that were under development. The “index.html” (Ex. C) confirms that these files were tested during development. Exhibits D, E and H demonstrate the availability of the supporting files called for by the demonstration plan. By comparison of Exhibits D, E and H to the demonstration plan (Ex. F), one sees that all of the files listed as desired for the demonstration were available before January 21, 1998. Our intended use of the “imdesc.xml” file was as an interface definition. (Ex. F, quoted above) In January 1998, we correctly understood that the imdesc.xml data structure would work for its intended purpose. Later versions of our interface definition data structure, such as the version that appears in our patent application, generally followed the document interface definition pattern of “imdesc.xml”. The interface definition version (Exhibit I) that was publicly disclosed at a conference on July 25, 1998, at the International Workshop on Component-based Electronic Commerce was held at the Fisher Center for Management & Information Technology, Haas School of Business, UC Berkeley also followed the same pattern.

17. The July 25 version, Slide 30, Exhibit I included:

```
<service>
<service.name>Order Service</service.name>
  <service.location>www.veosystems.com/order</service.location>
  <service.op>
    <service.op.name>Submit Order</service.op.name>
    <service.op.inputdoc>po.dtd</service.op.inputdoc>
    <service.op.outputdoc>poack.dtd</service.op.outputdoc>
  </service.op>
```

```

    < service.op>
      < service.op.name>Track Order</service.op.name>
      <service.op.inputdoc>request.track.dtd<service.op.inputdoc>
      <service.op.outputdoc>response.track.dtd<service.op.outputdoc>
    </service.op>
  </service>

```

A very similar version appears in the patent application on page 45, which I am informed and believe also appears as CD-ROM appendix LISTING 5, after reformatting of specification.

18. Our January 1998 understanding that this data structure would work for its intended purpose was subsequently validated by others, who adopted our approach to interface definition. For instance, the later W3 "note" dated March 15, 2001 describing WSDL version 1.1 followed the same document-based interface pattern. In Exhibit J, taken from <http://www.w3.org/TR/wsdl>, section 1.2, WSDL Document Example, we see a simplified version of our interface definition, which involves a single document exchange, as opposed to a series of related document exchanges:

```

<portType name="StockQuotePortType">
  <operation name="GetLastTradePrice">
    <input message="tns:GetLastTradePriceInput"/>
    <output message="tns:GetLastTradePriceOutput"/>
  </operation>
</portType>

```

19. Next, we compare the two versions of our data structure to the pending claims. Singularly and collectively, the versions of data structures, which were stored in memory accessible to at least one node on a network, included the characteristics and features described below.

Claim 1: In Exhibits G-I, both imdesc.xml and Slide 30 depict machine readable interface specifications. The imdesc.xml file typically is found in a file directory on a

machine readable storage media. The PowerPoint Slide 30 appears to have been taken from a computer file similar to imdesc.xml and pasted into the presentation. The archive for imdesc.xml and the PowerPoint presentation both were stored on machine readable storage media. Both data structures define an interface to a transaction process. The imdesc.xml defines an interface with several service “functions”. Slide 30 defines multiple service “operations”. In imdesc.xml, one of the input documents is an “order”; in Slide 30, there is a “po”, which is short for purchase order. In both transaction interface definitions, an acknowledgement is sent, an “ack” in the earlier version and a “poack” in the later version. The input document definitions are referenced by “order.dtd”, “invoiceo.dtd”, “paynoteo.dtd”, “po.dtd” and “request.track.dtd”, which are data type definition (dtd) files. The output document definitions are referenced by “ack.dtd”, “poack.dtd” and “response.track.dtd”. The January 1998 demonstration scenario (Ex. F, quoted above) makes it clear that these interface definitions were published to nodes on a network that might desire to invoke the functions for which interfaces were defined. The context of the slides in Exhibit I make it clear that this interface was hosted and accessible to a plurality of nodes on a network in the development environment from which it was borrowed for the PowerPoint presentation.

Claim 2: The data type specifications in the exemplary “order.dtd” (Ex. E) include at least one logical structure. Similarly, slides 25-29 (Ex. I) depict using logical structure building blocks to construct documents such as a purchase order (po.dtd). In general, a data type definition file (dtd) will include data type specifications for at least on logical data structure.

Claim 3: The data type specifications for the country field of the exemplary “order.dtd” (Ex. E) include at least one data structure mapping predefined sets of storage units for a particular logical structure in the definitions to respective entries in a list. The country code field is supported by a list in “codes.mod” (the codes module), which is incorporated into addresso.mod which, in turn, is part of order.dtd. With a bit of tracing from imdesc.xml, one can see where the reusable lists were part of the system. Turning to

the July 25, 1998 presentation, slides 27-28 similarly illustrate use of country codes in a purchase order, defined by the file "po.dtd".

Claim 4: In Exhibit D, the ENTITY "command" and the related ELEMENTs support registering definitions of documents and services in a repository in memory accessible to at least one node. The definitions of documents and services include logical structures and interpretation information for logical structures. In Exhibit I, Slide 23 depicts a service registration document in a registry accessible to discovery nodes on a network. Slide 31 indicates that CBL was already in use for demonstration applications, Project Seitai and GSA catalog interoperability. We found it very useful to post our interface definitions for programmers to rely on during development, such as the imdesc.xml interface definition.

Claim 5: Looking through files from the ingram/01 directory, which are reproduced in Exhibit H, one finds the data type definition corresponding to the sample interface definition imdesc.xml. Beginning with imdesc.xml on page 32 of 66, one sees reference to "imarkdsc.dtd" in the DOCTYPE statement of line six. This data type definition appears on page 29 of 66, and incorporates by reference the "isrvprim.mod" module that appears on page 54 of 66. The isrvprim.mod defines elements of imdesc.xml, including service.name, service.location.pointer, and service.function. Thus, the sample "imdesc.xml" machine readable specification of Exhibit G complies with the imarkdsc.dtd definition of an interface document including logical structures for storing identifiers (e.g., service.name and/or input doctype) and references to definitions (dtd's) of input and output documents. Similarly, in Exhibit I, slide 30, identifiers of particular transactions appear as "Submit Order" and "Track Order." The references to definitions of the input and output documents are dtd file names. The "Loose Coupling" via Shared Document Definitions slide 22 makes it clear that the definition in slide 30 is a document that complies with a dtd, as in the ingram/01 files.

Claim 6: Both the "imdesc.xml" Exhibit G and the Slide 30 in Exhibit I specify one or more transactions supported by the interface.

Claim 7: Both the “imdesc.xml” Exhibit G and the Slide 30 in Exhibit I include references to documents used in the particular transactions, specifying the document dtd’s by name instead of providing a copy of them.

Claims 8-12: The input and output documents in both versions are depicted as XML documents. Generally, XML documents compliant with the February 1998 recommendation can be parsed or unparsed data. XML documents may encode text characters and the text characters may provide a natural language word. XML documents generally include markup data to identify sets of storage units.

Claim 13: The files reprinted in Exhibits D, E and H from directories 072, 075 and ingram/01 store document types that can be used in a plurality of transactions. Commands identified above were used to register these document types in a repository. Slides 25-29 depict a library of document types stored in a repository. The dtd definitions for the documents used in the transactions are referenced by names. For instance, in “imdesc.xml” we see the “ack.dtd” as the output document for several service functions.

Claim 14: The collections of files from directories include “imarkprt.dtd” and “imarkdsc.dtd” for market participant information and market description information. These modules are called out in Slide 28 of Exhibit I by slightly different and more easily readable names, “markpart.dtd” and “markdesc.dtd”. Some of these document types include identifications of participant processes.

Claim 15: The dtd files referenced in “imdesc.xml” and Slide 30, e.g., order.dtd and ack.dtd, are recognizable as compliant with a standard Extensible Markup Language XML.

Claim 16: The file “imdesc.xml” and the code excerpt in Slide 30 are recognizable as compliant with a standard Extensible Markup Language XML.

Claim 61: In the course of creating “imdesc.xml” and Slide 30, the authors of those documents went through the process of defining a machine readable interface

definition including an input and output document. The versions of the transaction interface definition data structure were provided to network nodes that requested the definition data structures.

Claim 62: The data type specifications in the exemplary “order.dtd” include at least one logical structure. Similarly, slides 25-29 depict using logical structure building blocks to construct documents such as a purchase order (po.dtd). In general, a data type definition file (dtd) will include data type specifications for at least on logical data structure.

Claim 63: The data type specifications for the country field of the exemplary “order.dtd” (Ex. E) include at least one data structure mapping predefined sets of storage units for a particular logical structure in the definitions to respective entries in a list. The country code field is supported by a list in “codes.mod” (the codes module), which is incorporated into addresso.mod which, in turn, is part of order.dtd. With a bit of tracing from imdesc.xml, one can see where the reusable lists were part of the system. Use of country codes in addresses is a feature of CBL that extended the XML recommendation – the XML recommendation uses codes for language identification and not for addresses. See, W3C Recommendation, Extensible Markup Language (XML) 1.0, § 2.12 (Feb. 10, 1998) accessed at <http://www.w3.org/TR/1998/REC-xml-19980210#dt-app>. Turning to the July 25, 1998 presentation, slides 27-28 similarly illustrate use of country codes in a purchase order, defined by the file “po.dtd”.

Claim 64: Slide 23 (Ex. I) depicts a service registration document in a registry accessible to nodes on a network. Slide 31 indicates that CBL was already in use for demonstration applications, Project Seitai and GSA catalog interoperability. We found it very useful to post our interface definitions for programmers to rely on during development, such as the imdesc.xml interface definition. In the January 1998 and earlier materials (Exs. D, E and H), we have printed some libraries of logical structures that we were using then. The Slides indicate that the libraries of logical structures were still in use six months later, when the conference presentation was made in July 1998.

Claim 65: Looking through files from the ingram/01 directory (Ex. H) one finds how imdesc.xml has been defined. Beginning with imdesc.xml on page 32 of 66, one sees reference to "imarkdsc.dtd" in the DOCTYPE statement of line six. This data type definition appears on page 29 of 66, and incorporates by reference the "isrvprim.mod" module that appears on page 54 of 66. The isrvprim.mod defines elements of imdesc.xml, including service.name, service.location.pointer, and service.function. Thus, the sample "imdesc.xml" machine readable specification of Exhibit G complies with the imarkdsc.dtd definition of an interface document including logical structures for storing identifiers (e.g., service.name and/or input doctype) and references to definitions (dtd's) of input and output documents. Similarly, in Exhibit I, slide 30, identifiers of particular transactions appear as "Submit Order" and "Track Order." The references to definitions of the input and output documents are dtd file names. The "Loose Coupling" via Shared Document Definitions slide 22 makes it clear that the definition in slide 30 is a document that complies with a dtd, as in the ingram/01 files.

Claims 66-70: The input and output documents in both versions (code and presentation) are depicted as XML documents. Generally, XML documents compliant with the February 1998 recommendation can be parsed or unparsed data. XML documents may encode text characters and the text characters may provide a natural language word. XML documents generally include markup data to identify sets of storage units.

Claim 71: The dtd files referenced in "imdesc.xml" and Slide 30 are recognizable as compliant with a standard Extensible Markup Language XML.

Claim 72: Our development work in 1997-98 included applying parsers to input documents, generating one or more Java event signals in response to the logical structure of the input documents, and having event listener programs respond to the event signals.

20. I am informed and believe that Examiner Huynh has questioned whether versions of CBL prior to the publicly released version 1.1 worked. In fact, CBL worked for

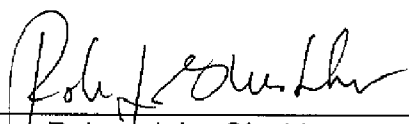
Application No.: 09/173,858

Docket No.: OIN 1004-1US

its intended purpose for many months and many versions prior to the September 1998 release of public version 1.1.

I declare under penalty of perjury of the laws of the United States of America that the foregoing is true and correct. I make this declaration with the understanding and knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 U.S.C. and that making willful false statements would jeopardize the validity of my application and any patents issuing thereon.

Executed this 17th day of July, 2008 in St Helena, CA.



Robert John Glushko

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Bart A. Meltzer

Application No.: 09/173,858

Confirmation No.: 4734

Filed: October 16, 1998

Art Unit: 2178

For: DOCUMENTS FOR COMMERCE IN
TRADING PARTNER NETWORKS AND
INTERFACE DEFINITIONS BASED ON THE
DOCUMENTS

Examiner: C. L. T. Huynh

**INVENTOR'S DECLARATION UNDER RULE 131 PROVING
REDUCTION TO PRACTICE ON OR BEFORE JANUARY 21, 1998**

I, Matthew Fuchs, declare as follows:

1. I am a former employee of CN Group, Veo and Commerce One and a named inventor. All of the statements made herein are my own personal knowledge or of my own opinion, based on my training and experience, except where stated on information and belief. I could competently testify thereto if called as a witness.

2. This declaration is given in support of the application entitled "Documents for Commerce in Trading Partner Networks and Interface Definitions Based on the Documents," U.S. patent application serial number 09/173,858, filed on October 18, 1998, and any other applications in which it might be submitted. In the course of preparation of this declaration, I looked at the documents listed in the attached list of exhibits, a set of patent claims, and this patent application. I am informed and believe that some of the exhibits were obtained from archives maintained by my former colleague Kevin Hughes.

3. In beginning in 1997 and continuing through 1998, I worked for CN Group, which became Veo and merged into Commerce One. In this declaration, "CN Group" and "Veo" are used interchangeably without any effort to track down the date or circumstance of the name change. Commerce One is mentioned in passing, as the events discussed in this declaration preceded the date on which merger of Veo into Commerce One became effective. Years ago, I received stock and/or stock options in Commerce One, which I sold at a profit before Commerce One declared bankruptcy.

4. I have been interested in the disposition of Commerce One's patents and patent applications, because they are very significant to business-to-business ecommerce. I am informed and believe that the Commerce One patents and patent applications were auctioned by a bankruptcy court and are now assigned to Open Invention Network ("OIN"), an organization formed by IBM, Phillips, Sony, Red Hat and Novell (and more recently joined by NEC), which has the web site www.openinventionnetwork.com. I am not involved in OIN, but I appreciate their stated dedication to acquire patents and offer them royalty free to promote Linux and spur innovation globally.

5. I have worked with OIN's counsel Mr. Beffel more than one occasion, to supply declarations that describe work I did as an inventor. We have had lunch and Mr. Beffel has promised to buy me and my wife dinner, in appreciation for my time.

6. Development of the technology disclosed in this patent application responded to a widely felt need for an object-oriented architectural framework for Internet commerce. The perceived need and efforts by IBM, Microsoft and others were described in an article by Tenenbaum, Jay M., Tripatinder S. Chowdhry and Kevin Hughes, "Eco System: An Internet Commerce Architecture" Computer May 1997: 48-55 ("Eco System article", submitted herewith, Exhibit A). The 1997 article proposed to use the conventional CORBA technology, which proved impractical.

7. CN Group shifted from developing a CORBA architecture and to a document-based transaction interface architecture months before W3C published XML as a

recommendation in February 1998. CN Group's change in approach is explained in a second article by Glushko, Robert J., Jay M. Tenenbaum, Bart Meltzer, "An XML Framework for Agent-based E-commerce" Communications of the ACM, Vol. 42, No. 3, pp.106-109 & 111-114 (March 1999) ("XML Framework", submitted herewith, Exhibit B) that published after this application was filed. Note that several of the figures in the XML Framework article resemble figures used in the patent application.

8. Before this patent application was filed, we developed multiple versions of data structures that established how to define a document-based transaction interface architecture. We could see that these data structures, when placed in memory, would define an interface to a transaction using one or more input documents and one or more output documents. The interface defined by these data structures was practical and workable, in contrast to the CORBA architecture proposed in the 1997 article. We understood that these data structures also could drive a code generating process that would generate Java class definitions for internal data manipulation and translators between XML and Java for marshalling and unmarshalling data between external and internal formats. Unlike CORBA, the document-based transaction interface architecture involved programs communicating with one another in a human readable external format, while using a machine-friendly internal format. So-called marshalling and unmarshalling components translated between XML and Java.

9. This declaration presents two of the data structure versions developed to define a document-based transaction interface architecture, one for a demonstration and one presented during a conference. The earlier version showed a multi-stage transaction and specified several document exchanges as parts of the transaction. It is dated January 3, 1998, which is well in advance of a demonstration that was scheduled for January 21, 1998 (Exhibit F), which demonstration may have been postponed. The conference version was presented on July 25, 1998 (Exhibit I, slide 30).

10. The XML Framework article and this patent application explain how CBL was used to standardize interpretation data for XML documents. The resulting XML documents

are useful as input and output documents referenced in a data structure that defines a document-based transaction interface architecture. However, a document-based transaction interface need not be implemented using CBL and CBL can be used in other ways that do not involve using a document-based transaction interface.

11. Submitted with this declaration are files from three versions of CBL and infrastructure supporting transactions, which I am informed and believe were found in directories named 072, 075 and "ingram/01," and which bore file date stamps and internal documentary dates before January 21, 1998.

(a) An index.html file (Ex. C) from directory 072 lists many modules as included in version 0.7.2, before the end of 1997. The index.html file also provides a limited description of module testing and validation that had been completed before the end of 1997.

(b) I am informed and believe that reprinted files in Exhibit D come from the directory 072, even though some of them have earlier version numbers, such as catentry.dtd version 0.6 and cblcat.dtd version 0.6.1..

(c) The modules in directory 075 are internally documented as belonging to version 0.8, but I am informed and believe that another version of the modules appears in a directory 080, with later dates. I am informed and believe that reprinted files in Exhibit E come from directory 075.

(d) I am informed and believe that reprinted files in Exhibit H come from directory ingram/01.

(e) Before January 21, 1998, files and modules from the three versions of CBL and infrastructure supporting transactions represented in Exhibits C-E and H were available for use in a demonstration.

12. Exhibit D consists of files dated from November 2 through December 5, 1997 from the directory 072. It includes, in the file "command.dtd" dated December 5, 1997, definitions of an ENTITY and ELEMENTs used for registering documents and services in a registry. The ENTITY is named "command" and the ELEMENTs include "command.set", "register.document" and "register.service". Registering documents and services was part of the infrastructure that we developed for using a choreographed exchange of documents to conduct a transaction. That is, as of December 5, 1997, Exhibit D already includes the basic infrastructure defining the registration of documents and services in a registry.

13. Submitted as Exhibit F is a plan for preparing for a demonstration to Ingram Micro, entitled "Requirements and Tasks for the January Demo, (Updated 1/6/98 by Kenneth)". The plan called for "1-3 clients which run in a browser" and "1 server process". "All information exchange is done via CBL/XML streams ..." The demonstration scenario is described as follows, from pages 4 and 6 of the plan:

"1. Market Description. Ingram creates **imdesc.xml** and publicizes it. imdesc.xml refers to ingram.xml.

"2. IBM, DEC, COMPUSA, and Fry's send market participant info to Ingram so they can be approved and listed in the market's directories.

"ibm.xml, dec.xml, compu.xml, frys.xml

"Ingram acks and registers the documents.

"3. IBM and DEC send product descriptions to Ingram, think.xml and hinote.xml. Ingram acks and from those product descriptions produces its own catalogue entries for them. Those cat entries are TBS. (May need multiple files for the multiple configs of these products.)

"4. COMPUSA and Fry's inquire about laptops that Ingram has. NEED DETAILS. TBS: the information request docs and the responses.

"5. We pretend that both COMPUSA and Fry's buy some of each laptop.

"6. IBM and DEC inquire about laptops that Ingram has and has sold (= laptops in the channel?). TBS: the information request docs and the responses.

“(More details)

“1) Setting up the "master node" in the channel --defining Ingram's role and the services it will be providing (e.g registration, integrated catalogs, part number mapping, handling queries about prices and inventories, ordering)

“2) Registering companies to participate in the channel as either (a) manufacturers or (b) resellers [this means describing themselves (core metadata), their services (e.g., their catalog schemas), their forms (their business document schemas)]

“3) The reseller's view of the channel:

a) show me the integrated catalog

b) here's a part number: what is everyone else calling it

b) show me my price list

“4) The manufacturer's view of the channel:

a) show me (my items in) the channel

b) here's a part number: what is everyone else calling it

c) what prices are being charged for this part”

The plan for demonstration, on pages 4-5, lists many components, including “.mod” modules and “.dtd” data type definitions for XML documents. Components with names matching all of the names listed in the plan appear in the attached reprints from the 072, 075 and ingram/01 directories. Note that the scenario point # 1 is, “Ingram creates imdesc.xml and publicizes it.”

14. The file “imdesc.xml” (Exhibit G), which I am informed and believe comes from the “ingram/01” directory, is one version of a data structure that defines an interface to a transaction that includes multiple input documents and output documents. The file is written in XML and points to “.dtd” definitions of input and output documents for various services related to “Ordering and Fulfillment”. The file includes, in part, the text:

```
<!-- imdesc.xml Version: 0.1 -->
<!-- Purpose: marketplace description for Ingram Micro demo -->
<!-- Terry Allen 2 Jan 1998 -->
<!-- Copyright 1998 CNgrou, Inc. -->
<?xml version="1.0"?>
<!DOCTYPE market.description SYSTEM "imarkdsc.dtd">
<market.description>
```

```

<market.name>Ingram Micro Online Sales Network
</market.name>
<market.operator>
<market.operator.name>Ingram Micro Inc.
</market.operator.name>
<market.participant.info.pointer>
  <xll.locator urlink="ingram.xml">Ingram's Market Participant Info
  </xll.locator>
</market.participant.info.pointer>
</market.operator>
<service.set>
  <service>
    <service.name>Ordering and Fulfillment
    </service.name>
    <service.function.sequence>
      <service.function>
        <doctype from.party="any" to.party="ingram">order.dtd</doctype>
        <doctype from.party="ingram" to.party="any">ack.dtd</doctype>
      </service.function>
      <service.function>
        <doctype from.party="ingram" to.party="any">invoiceo.dtd</doctype>
        <doctype from.party="any" to.party="ingram">ack.dtd</doctype>
      </service.function>
      <service.function>
        <doctype from.party="ingram" to.party="any">shipnote.dtd</doctype>
      </service.function>
      <service.function>
        <doctype from.party="any" to.party="ingram">paynoteo.dtd</doctype>
        <doctype from.party="ingram" to.party="any">ack.dtd</doctype>
      </service.function>
    </service.function.sequence>
  </service>
</service.set> ...

```

The bold facing is added for emphasis. The internal documentation date of this file is January 2, 1998. I am informed and believe that the file date stamp indicates that the file was last changed on January 4, 1998 at 3:47 a.m. Both dates are well in advance of the scheduled demonstration on January 21, 1998.

15. The imdesc.xml file, in the excerpt above, describes service functions for "Ordering and Fulfillment" that are registered to be accessible through Ingram's URL. One

service receives an order and acknowledges it. Another generates an invoice and expects an acknowledgment. A third generates a shipping notice, without expecting an acknowledgment. Another receives a payment notice and acknowledges it. Each of these services is accessible through an interface that accepts an XML input document that conforms to a “.dtd” file and, typically, returns an XML acknowledgement document.

16. Before January 21, 1998, we had sufficiently worked with imdesc.xml to recognize and understand that it would serve its intended purpose of defining a document-based transaction interface that can include a series of related document exchanges. This file was one of many that were under development. The “index.html” (Ex. C) confirms that these files were tested during development. Exhibits D, E and H demonstrate the availability of the supporting files called for by the demonstration plan. By comparison of Exhibits D, E and H to the demonstration plan (Ex. F), one sees that all of the files listed as desired for the demonstration were available before January 21, 1998. Our intended use of the “imdesc.xml” file was as an interface definition. (Ex. F, quoted above) In January 1998, we correctly understood that the imdesc.xml data structure would work for its intended purpose. Later versions of our interface definition data structure, such as the version that appears in our patent application, generally followed the document interface definition pattern of “imdesc.xml”. The interface definition version (Exhibit I) that was publicly disclosed at a conference on July 25, 1998, at the International Workshop on Component-based Electronic Commerce was held at the Fisher Center for Management & Information Technology, Haas School of Business, UC Berkeley also followed the same pattern.

17. The July 25 version, Slide 30, Exhibit I included:

```
<service>
<service.name>Order Service</service.name>
  <service.location>www.veosystems.com/order</service.location>
  <service.op>
    <service.op.name>Submit Order</service.op.name>
    <service.op.inputdoc>po.dtd</service.op.inputdoc>
    <service.op.outputdoc>poack.dtd</service.op.outputdoc>
  </service.op>
```

```

    < service.op>
      < service.op.name>Track Order</service.op.name>
      <service.op.inputdoc>request.track.dtd<service.op.inputdoc>
      <service.op.outputdoc>response.track.dtd<service.op.outputdoc>
    </service.op>
  </service>

```

A very similar version appears in the patent application on page 45, which I am informed and believe also appears as CD-ROM appendix LISTING 5, after reformatting of specification.

18. Our January 1998 understanding that this data structure would work for its intended purpose was subsequently validated by others, who adopted our approach to interface definition. For instance, the later W3 "note" dated March 15, 2001 describing WSDL version 1.1 followed the same document-based interface pattern. In Exhibit J, taken from <http://www.w3.org/TR/wsdl>, section 1.2, WSDL Document Example, we see a simplified version of our interface definition, which involves a single document exchange, as opposed to a series of related document exchanges:

```

<portType name="StockQuotePortType">
  <operation name="GetLastTradePrice">
    <input message="tns:GetLastTradePriceInput"/>
    <output message="tns:GetLastTradePriceOutput"/>
  </operation>
</portType>

```

19. Next, we compare the two versions of our data structure to the pending claims. Singularly and collectively, the versions of data structures, which were stored in memory accessible to at least one node on a network, included the characteristics and features described below.

Claim 1: In Exhibits G-I, both imdesc.xml and Slide 30 depict machine readable interface specifications. The imdesc.xml file typically is found in a file directory on a

machine readable storage media. The PowerPoint Slide 30 appears to have been taken from a computer file similar to imdesc.xml and pasted into the presentation. The archive for imdesc.xml and the PowerPoint presentation both were stored on machine readable storage media. Both data structures define an interface to a transaction process. The imdesc.xml defines an interface with several service “functions”. Slide 30 defines multiple service “operations”. In imdesc.xml, one of the input documents is an “order”; in Slide 30, there is a “po”, which is short for purchase order. In both transaction interface definitions, an acknowledgement is sent, an “ack” in the earlier version and a “poack” in the later version. The input document definitions are referenced by “order.dtd”, “invoiceo.dtd”, “paynoteo.dtd”, “po.dtd” and “request.track.dtd”, which are data type definition (dtd) files. The output document definitions are referenced by “ack.dtd”, “poack.dtd” and “response.track.dtd”. The January 1998 demonstration scenario (Ex. F, quoted above) makes it clear that these interface definitions were published to nodes on a network that might desire to invoke the functions for which interfaces were defined. The context of the slides in Exhibit I makes it clear that this interface was hosted and accessible to a plurality of nodes on a network in the development environment from which it was borrowed for the PowerPoint presentation.

Claim 2: The data type specifications in the exemplary “order.dtd” (Ex. E) include at least one logical structure. Similarly, slides 25-29 (Ex. I) depict using logical structure building blocks to construct documents such as a purchase order (po.dtd). In general, a data type definition file (dtd) will include data type specifications for at least on logical data structure.

Claim 3: The data type specifications for the country field of the exemplary “order.dtd” (Ex. E) include at least one data structure mapping predefined sets of storage units for a particular logical structure in the definitions to respective entries in a list. The country code field is supported by a list in “codes.mod” (the codes module), which is incorporated into addresso.mod which, in turn, is part of order.dtd. With a bit of tracing from imdesc.xml, one can see where the reusable lists were part of the system. Turning to

the July 25, 1998 presentation, slides 27-28 similarly illustrate use of country codes in a purchase order, defined by the file "po.dtd".

Claim 4: In Exhibit D, the ENTITY "command" and the related ELEMENTs support registering definitions of documents and services in a repository in memory accessible to at least one node. The definitions of documents and services include logical structures and interpretation information for logical structures. In Exhibit I, Slide 23 depicts a service registration document in a registry accessible to discovery nodes on a network. Slide 31 indicates that CBL was already in use for demonstration applications, Project Seitai and GSA catalog interoperability. We found it very useful to post our interface definitions for programmers to rely on during development, such as the imdesc.xml interface definition.

Claim 5: Looking through files from the ingram/01 directory, which are reproduced in Exhibit H, one finds the data type definition corresponding to the sample interface definition imdesc.xml. Beginning with imdesc.xml on page 32 of 66, one sees reference to "imarkdsc.dtd" in the DOCTYPE statement of line six. This data type definition appears on page 29 of 66, and incorporates by reference the "isrvprim.mod" module that appears on page 54 of 66. The isrvprim.mod defines elements of imdesc.xml, including service.name, service.location.pointer, and service.function. Thus, the sample "imdesc.xml" machine readable specification of Exhibit G complies with the imarkdsc.dtd definition of an interface document including logical structures for storing identifiers (e.g., service.name and/or input doctype) and references to definitions (dtd's) of input and output documents. Similarly, in Exhibit I, slide 30, identifiers of particular transactions appear as "Submit Order" and "Track Order." The references to definitions of the input and output documents are dtd file names. The "Loose Coupling" via Shared Document Definitions slide 22 makes it clear that the definition in slide 30 is a document that complies with a dtd, as in the ingram/01 files.

Claim 6: Both the "imdesc.xml" Exhibit G and the Slide 30 in Exhibit I specify one or more transactions supported by the interface.

Claim 7: Both the “imdesc.xml” Exhibit G and the Slide 30 in Exhibit I include references to documents used in the particular transactions, specifying the document dtd’s by name instead of providing a copy of them.

Claims 8-12: The input and output documents in both versions are depicted as XML documents. Generally, XML documents compliant with the February 1998 recommendation can be parsed or unparsed data. XML documents may encode text characters and the text characters may provide a natural language word. XML documents generally include markup data to identify sets of storage units.

Claim 13: The files reprinted in Exhibits D, E and H from directories 072, 075 and ingram/01 store document types that can be used in a plurality of transactions. Commands identified above were used to register these document types in a repository. Slides 25-29 depict a library of document types stored in a repository. The dtd definitions for the documents used in the transactions are referenced by names. For instance, in “imdesc.xml” we see the “ack.dtd” as the output document for several service functions.

Claim 14: The collections of files from directories include “imarkprt.dtd” and “imarkdsc.dtd” for market participant information and market description information. These modules are called out in Slide 28 of Exhibit I by slightly different and more easily readable names, “markpart.dtd” and “markdesc.dtd”. Some of these document types include identifications of participant processes.

Claim 15: The dtd files referenced in “imdesc.xml” and Slide 30, e.g., order.dtd and ack.dtd, are recognizable as compliant with a standard Extensible Markup Language XML.

Claim 16: The file “imdesc.xml” and the code excerpt in Slide 30 are recognizable as compliant with a standard Extensible Markup Language XML.

Claim 61: In the course of creating “imdesc.xml” and Slide 30, the authors of those documents went through the process of defining a machine readable interface

definition including an input and output document. The versions of the transaction interface definition data structure were provided to network nodes that requested the definition data structures.

Claim 62: The data type specifications in the exemplary “order.dtd” include at least one logical structure. Similarly, slides 25-29 depict using logical structure building blocks to construct documents such as a purchase order (po.dtd). In general, a data type definition file (dtd) will include data type specifications for at least on logical data structure.

Claim 63: The data type specifications for the country field of the exemplary “order.dtd” (Ex. E) include at least one data structure mapping predefined sets of storage units for a particular logical structure in the definitions to respective entries in a list. The country code field is supported by a list in “codes.mod” (the codes module), which is incorporated into addresso.mod which, in turn, is part of order.dtd. With a bit of tracing from imdesc.xml, one can see where the reusable lists were part of the system. Use of country codes in addresses is a feature of CBL that extended the XML recommendation – the XML recommendation uses codes for language identification and not for addresses. See, W3C Recommendation, Extensible Markup Language (XML) 1.0, § 2.12 (Feb. 10, 1998) accessed at <http://www.w3.org/TR/1998/REC-xml-19980210#dt-app>. Turning to the July 25, 1998 presentation, slides 27-28 similarly illustrate use of country codes in a purchase order, defined by the file “po.dtd”.

Claim 64: Slide 23 (Ex. I) depicts a service registration document in a registry accessible to nodes on a network. Slide 31 indicates that CBL was already in use for demonstration applications, Project Seitai and GSA catalog interoperability. We found it very useful to post our interface definitions for programmers to rely on during development, such as the imdesc.xml interface definition. In the January 1998 and earlier materials (Exs. D, E and H), we have printed some libraries of logical structures that we were using then. The Slides indicate that the libraries of logical structures were still in use six months later, when the conference presentation was made in July 1998.

Claim 65: Looking through files from the ingram/01 directory (Ex. H) one finds how imdesc.xml has been defined. Beginning with imdesc.xml on page 32 of 66, one sees reference to "imarkdsc.dtd" in the DOCTYPE statement of line six. This data type definition appears on page 29 of 66, and incorporates by reference the "isrvprim.mod" module that appears on page 54 of 66. The isrvprim.mod defines elements of imdesc.xml, including service.name, service.location.pointer, and service.function. Thus, the sample "imdesc.xml" machine readable specification of Exhibit G complies with the imarkdsc.dtd definition of an interface document including logical structures for storing identifiers (e.g., service.name and/or input doctype) and references to definitions (dtd's) of input and output documents. Similarly, in Exhibit I, slide 30, identifiers of particular transactions appear as "Submit Order" and "Track Order." The references to definitions of the input and output documents are dtd file names. The "Loose Coupling" via Shared Document Definitions slide 22 makes it clear that the definition in slide 30 is a document that complies with a dtd, as in the ingram/01 files.

Claims 66-70: The input and output documents in both versions (code and presentation) are depicted as XML documents. Generally, XML documents compliant with the February 1998 recommendation can be parsed or unparsed data. XML documents may encode text characters and the text characters may provide a natural language word. XML documents generally include markup data to identify sets of storage units.

Claim 71: The dtd files referenced in "imdesc.xml" and Slide 30 are recognizable as compliant with a standard Extensible Markup Language XML.

Claim 72: Our development work in 1997-98 included applying parsers to input documents, generating one or more Java event signals in response to the logical structure of the input documents, and having event listener programs respond to the event signals.

20. I am informed and believe that Examiner Huynh has questioned whether versions of CBL prior to the publicly released version 1.1 worked. In fact, CBL worked for

Application No.: 09/173,858

Docket No.: OIN 1004-1US

its intended purpose for many months and many versions prior to the September 1998 release of public version 1.1.

I declare under penalty of perjury of the laws of the United States of America that the foregoing is true and correct. I make this declaration with the understanding and knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 U.S.C. and that making willful false statements would jeopardize the validity of my application and any patents issuing thereon.

Executed this 21th day of July, 2008 in Palo Alto, CA


Matthew Fuchs.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Bart A. Meltzer

Application No.: 09/173,858

Confirmation No.: 4734

Filed: October 16, 1998

Art Unit: 2178

For: DOCUMENTS FOR COMMERCE IN
TRADING PARTNER NETWORKS AND
INTERFACE DEFINITIONS BASED ON THE
DOCUMENTS

Examiner: C. L. T. Huynh

**NON-INVENTOR'S DECLARATION UNDER RULE 131 PROVING
REDUCTION TO PRACTICE ON OR BEFORE JANUARY 21, 1998**

I I, Kevin Hughes, declare as follows:

1. I am a former employee of CommerceNet, CN Group, Veo and Commerce One and a named author of one of the articles discussed in this declaration. One of my career accomplishments was induction in 1994 into the World Wide Web Hall of Fame, with a few others including Mark Andreessen of Netscape. Another is being a recipient of the University of Hawaii's Distinguished Alumni Award. All of the statements made herein are my own personal knowledge or of my own opinion, based on my training and experience, except where stated on information and belief. I could competently testify thereto if called as a witness.

2. This declaration is given in support of the application entitled "Documents for Commerce in Trading Partner Networks and Interface Definitions Based on the Documents," U.S. patent application serial number 09/173,858, filed on October 18, 1998, and any other applications in which it might be submitted. In the course of preparation of this declaration, I looked at the documents listed in the attached list of exhibits, a set of

patent claims, and this patent application. I recognize many of the exhibits as reprints from archives that I maintained and provided, as described below.

3. In 1997, I worked for CommerceNet, a non-profit organization. Some of the people involved in CommerceNet, including myself, also worked for CN Group, which became Veo and merged into Commerce One. In this declaration, "CN Group" and "Veo" are used interchangeably without any effort to track down the date or circumstance of the name change. Commerce One is mentioned in passing, as the events discussed in this declaration preceded the date on which merger of Veo into Commerce One became effective. I worked with the inventors named in this application and have stayed in touch with some of them. Years ago, I received stock and/or stock options in Commerce One, which I sold at a profit before Commerce One declared bankruptcy.

4. I have been interested in the disposition of Commerce One's patents and patent applications, because they are very significant to business-to-business ecommerce. I am informed and believe that the Commerce One patents and patent applications were auctioned by a bankruptcy court and are now assigned to Open Invention Network ("OIN"), an organization formed by IBM, Phillips, Sony, Red Hat and Novell (and more recently joined by NEC), which has the web site www.openinventionnetwork.com. I am not involved in OIN, but I appreciate their stated dedication to acquire patents and offer them royalty free to promote Linux and spur innovation globally.

5. In fall of 2007, OIN's counsel Mr. Beffel contacted me and inquired about the existence of archival files from CN Group that would reflect development work in 1997-98. I located and provided Mr. Beffel with some 1997-98 and later files. The files that I provided were true and correct copies of files to which I had access during the regular course of business, while working with CommerceNet, CN Group, Veo and/or Commerce One. I created some of the files myself. I worked with files that others created. I have compared the files in the accompanying exhibits that purport to be reprinted copies with my archives of electronic files. The reprinted files in the accompanying exhibits are authentic reprints of files developed in the ordinary course of business by CN Group and Veo and maintained in

my archives. I have not been compensated for my work on this declaration. I have spoken with but not met Mr. Beffel.

6. Development of the technology disclosed in this patent application responded to a widely felt need for an object-oriented architectural framework for Internet commerce. The perceived need and efforts by IBM, Microsoft and others were described in an article by Tenenbaum, Jay M., Tripatinder S. Chowdhry and Kevin Hughes, "Eco System: An Internet Commerce Architecture" Computer May 1997: 48-55 ("Eco System article", submitted herewith, Exhibit A). The 1997 article proposed to use the conventional CORBA technology, which proved impractical.

7. CN Group shifted from developing a CORBA architecture and to a document-based transaction interface architecture months before W3C published XML as a recommendation in February 1998. CN Group's change in approach is explained in a second article by Glushko, Robert J., Jay M. Tenenbaum, Bart Meltzer, "An XML Framework for Agent-based E-commerce" Communications of the ACM, Vol. 42, No. 3, pp.106-109 & 111-114 (March 1999) ("XML Framework", submitted herewith, Exhibit B) that published after this application was filed. Note that several of the figures in the XML Framework article resemble figures used in the patent application.

8. Before this patent application was filed, we developed multiple versions of data structures that established how to define a document-based transaction interface architecture. We could see that these data structures, when placed in memory, would define an interface to a transaction using one or more input documents and one or more output documents. The interface defined by these data structures was practical and workable, in contrast to the CORBA architecture proposed in the 1997 article. We understood that these data structures also could drive a code generating process that would generate Java class definitions for internal data manipulation and translators between XML and Java for marshalling and unmarshalling data between external and internal formats. Unlike CORBA, the document-based transaction interface architecture involved programs communicating with one another in a human readable external format,

while using a machine-friendly internal format. So-called marshalling and unmarshalling components translated between XML and Java.

9. This declaration presents two of the data structure versions developed to define a document-based transaction interface architecture, one for a demonstration and one presented during a conference. The earlier version showed a multi-stage transaction and specified several document exchanges as parts of the transaction. It is dated January 3, 1998, which is well in advance of a demonstration that was scheduled for January 21, 1998 (Exhibit F), which demonstration may have been postponed. The conference version was presented on July 25, 1998 (Exhibit I, slide 30).

10. The XML Framework article and this patent application explain how CBL was used to standardize interpretation data for XML documents. The resulting XML documents are useful as input and output documents referenced in a data structure that defines a document-based transaction interface architecture. However, CBL need not be implemented using a document-based transaction interface.

11. Submitted with this declaration are files from three versions of CBL and infrastructure supporting transactions, which I am informed and believe were found in directories named 072, 075 and "ingram/01," and which bore file date stamps and internal documentary dates before January 21, 1998.

(a) An index.html file (Ex. C) from directory 072 lists many modules as included in version 0.7.2, before the end of 1997. The index.html file also provides a limited description of module testing and validation that had been completed before the end of 1997.

(b) I am informed and believe that reprinted files in Exhibit D come from the directory 072, even though some of them have earlier version numbers, such as catentry.dtd version 0.6 and cblcat.dtd version 0.6.1.

(c) The modules in directory 075 are internally documented as belonging to version 0.8, but I am informed and believe that another version of the modules appears in a directory 080, with later dates. I am informed and believe that reprinted files in Exhibit E come from directory 075.

(d) I am informed and believe that reprinted files in Exhibit H come from directory ingram/01.

(e) Before January 21, 1998, files and modules from the three versions of CBL and infrastructure supporting transactions represented in Exhibits C-E and H were available for use in a demonstration.

12. Exhibit D consists of files dated from November 2 through December 5, 1997 from the directory 072. It includes, in the file "command.dtd" dated December 5, 1997, definitions of an ENTITY and ELEMENTs used for registering documents and services in a registry. The ENTITY is named "command" and the ELEMENTs include "command.set", "register.document" and "register.service". Registering documents and services was part of the infrastructure that we developed for using a choreographed exchange of documents to conduct a transaction. That is, as of December 5, 1997, Exhibit D already includes the basic infrastructure defining the registration of documents and services in a registry.

13. Submitted as Exhibit F is a plan for preparing for a demonstration to Ingram Micro, entitled "Requirements and Tasks for the January Demo, (Updated 1/6/98 by Kenneth)". The plan called for "1-3 clients which run in a browser" and "1 server process". "All information exchange is done via CBL/XML streams ..." The demonstration scenario is described as follows, from pages 4 and 6 of the plan:

"1. Market Description. Ingram creates **imdesc.xml** and publicizes it. imdesc.xml refers to ingram.xml.

"2. IBM, DEC, COMPUSA, and Fry's send market participant info to Ingram so they can be approved and listed in the market's directories.

"ibm.xml, dec.xml, compu.xml, frys.xml

"Ingram acks and registers the documents.

"3. IBM and DEC send product descriptions to Ingram, think.xml and hinote.xml. Ingram acks and from those product descriptions produces its own catalogue entries for them. Those cat entries are TBS. (May need multiple files for the multiple configs of these products.)

"4. COMPUSA and Fry's inquire about laptops that Ingram has. NEED DETAILS. TBS: the information request docs and the responses.

"5. We pretend that both COMPUSA and Fry's buy some of each laptop.

"6. IBM and DEC inquire about laptops that Ingram has and has sold (= laptops in the channel?). TBS: the information request docs and the responses.

"(More details)

"1) Setting up the "master node" in the channel --defining Ingram's role and the services it will be providing (e.g registration, integrated catalogs, part number mapping, handling queries about prices and inventories, ordering)

"2) Registering companies to participate in the channel as either (a) manufacturers or (b) resellers [this means describing themselves (core metadata), their services (e.g., their catalog schemas), their forms (their business document schemas)]

"3) The reseller's view of the channel:

- a) show me the integrated catalog
- b) here's a part number: what is everyone else calling it
- b) show me my price list

"4) The manufacturer's view of the channel:

- a) show me (my items in) the channel
- b) here's a part number: what is everyone else calling it
- c) what prices are being charged for this part"

The plan for demonstration, on pages 4-5, lists many components, including ".mod" modules and ".dtd" data type definitions for XML documents. Components with names matching all of the names listed in the plan appear in the attached reprints from the 072, 075 and ingram/01 directories. Note that the scenario point # 1 is, "Ingram creates imdesc.xml and publicizes it."

14. The file "imdesc.xml" (Exhibit G), which I am informed and believe comes from the "ingram/01" directory, is one version of a data structure that defines an interface to a transaction that includes multiple input documents and output documents. The file is written in XML and points to ".dtd" definitions of input and output documents for various services related to "Ordering and Fulfillment". The file includes, in part, the text:

```
<!-- imdesc.xml Version: 0.1 -->
<!-- Purpose: marketplace description for Ingram Micro demo -->
<!-- Terry Allen 2 Jan 1998 -->
<!-- Copyright 1998 CNgrou, Inc. -->
<?xml version="1.0"?>
<!DOCTYPE market.description SYSTEM "imarkdsc.dtd">
<market.description>
  <market.name>Ingram Micro Online Sales Network
</market.name>
  <market.operator>
    <market.operator.name>Ingram Micro Inc.
  </market.operator.name>
  <market.participant.info.pointer>
    <xll.locator urlink="ingram.xml">Ingram's Market Participant Info
  </xll.locator>
  </market.participant.info.pointer>
</market.operator>
<service.set>
  <service>
    <service.name>Ordering and Fulfillment
  </service.name>
  <service.function.sequence>
    <service.function>
      <doctype from.party="any" to.party="ingram">order.dtd</doctype>
      <doctype from.party="ingram" to.party="any">ack.dtd</doctype>
    </service.function>
    <service.function>
      <doctype from.party="ingram" to.party="any">invoiceo.dtd</doctype>
      <doctype from.party="any" to.party="ingram">ack.dtd</doctype>
    </service.function>
    <service.function>
      <doctype from.party="ingram" to.party="any">shipnote.dtd</doctype>
    </service.function>
    <service.function>
      <doctype from.party="any" to.party="ingram">paynoteo.dtd</doctype>
      <doctype from.party="ingram" to.party="any">ack.dtd</doctype>
    </service.function>
  </service>
</service.set>
```

```
</service.function>
</service.function.sequence>
</service>
<service> ...
```

The bold facing is added for emphasis. The internal documentation date of this file is January 2, 1998. I am informed and believe that the file date stamp indicates that the file was last opened on January 4, 1998 at 3:47 a.m. Both dates are well in advance of the scheduled demonstration on January 21, 1998.

15. The imdesc.xml file, in the excerpt above, describes service functions for "Ordering and Fulfillment" that are registered to be accessible through Ingram's URL. One service receives an order and acknowledges it. Another generates an invoice and expects an acknowledgment. A third generates a shipping notice, without expecting an acknowledgement. Another receives a payment notice and acknowledges it. Each of these services is accessible through an interface that accepts an XML input document that conforms to a ".dtd" file and, typically, returns an XML acknowledgement document.

16. Before January 21, 1998, we had sufficiently worked with imdesc.xml to recognize and understand that it would serve its intended purpose of defining a document-based transaction interface that can include a series of related document exchanges. This file was one of many that were under development. The "index.html" (Ex. C) confirms that these files were tested during development. Exhibits D, E and H demonstrate the availability of the supporting files called for by the demonstration plan. By comparison of Exhibits D, E and H to the demonstration plan (Ex. F), one sees that all of the files listed as desired for the demonstration were available before January 21, 1998. Our intended use of the "imdesc.xml" file was as an interface definition. (Ex. F, quoted above) In January 1998, we correctly understood that the imdesc.xml data structure would work for its intended purpose. Later versions of our interface definition data structure, such as the version that appears in our patent application, generally followed the document interface definition pattern of "imdesc.xml". The interface definition version (Exhibit I) that was publicly disclosed at a conference on July 25, 1998, at the International Workshop on Component-

based Electronic Commerce was held at the Fisher Center for Management & Information Technology, Haas School of Business, UC Berkeley also followed the same pattern.

17. The July 25 version, Slide 30, Exhibit I included:

```
<service>
  <service.name>Order Service</service.name>
  <service.location>www.veosystems.com/order</service.location>
  <service.op>
    <service.op.name>Submit Order</service.op.name>
    <service.op.inputdoc>po.dtd</service.op.inputdoc>
    <service.op.outputdoc>poack.dtd</service.op.outputdoc>
  </service.op>
  < service.op>
    < service.op.name>Track Order</service.op.name>
    <service.op.inputdoc>request.track.dtd<service.op.inputdoc>
    <service.op.outputdoc>response.track.dtd<service.op.outputdoc>
  </service.op>
</service>
```

A very similar version appears in the patent application on page 45, which I am informed and believe also appears as CD-ROM appendix LISTING 5, after reformatting of specification.

18. Our January 1998 understanding that this data structure would work for its intended purpose was subsequently validated by others, who adopted our approach to interface definition. For instance, the later W3 "note" dated March 15, 2001 describing WSDL version 1.1 followed the same document-based interface pattern. In Exhibit J, taken from <http://www.w3.org/TR/wsdl>, section 1.2, WSDL Document Example, we see a simplified version of our interface definition, which involves a single document exchange, as opposed to a series of related document exchanges:

```
<portType name="StockQuotePortType">
  <operation name="GetLastTradePrice">
    <input message="tns:GetLastTradePriceInput"/>
    <output message="tns:GetLastTradePriceOutput"/>
  </operation>
</portType>
```

</operation>
</portType>

19. Next, we compare the two versions of our data structure to the pending claims. Singularly and collectively, the versions of data structures, which were stored in memory accessible to at least one node on a network, included the characteristics and features described below.

Claim 1: In Exhibits G-I, both imdesc.xml and Slide 30 depict machine readable interface specifications. The imdesc.xml file typically is found in a file directory on a machine readable storage media. The Powerpoint Slide 30 appears to have been taken from a computer file similar to imdesc.xml and pasted into the presentation. The archive for imdesc.xml and the Powerpoint presentation both were stored on machine readable storage media. Both data structures define an interface to a transaction process. The imdesc.xml defines an interface with several service “functions”. Slide 30 defines multiple service “operations”. In imdesc.xml, one of the input documents is an “order”; in Slide 30, there is a “po”, which is short for purchase order. In both transaction interface definitions, an acknowledgement is sent, an “ack” in the earlier version and a “poack” in the later version. The input document definitions are referenced by “order.dtd”, “invoiceo.dtd”, “paynoteo.dtd”, “po.dtd” and “request.track.dtd”, which are data type definition (dtd) files. The output document definitions are referenced by “ack.dtd”, “poack.dtd” and “response.track.dtd”. The January 1998 demonstration scenario (Ex. F, quoted above) makes it clear that these interface definitions were published to nodes on a network that might desire to invoke the functions for which interfaces were defined. The context of the slides and other remarks by Glushko, including ongoing demonstration projects, make it clear that this interface was hosted and accessible to a plurality of nodes on a network in the development environment from which it was borrowed for the Powerpoint presentation.

Claim 2: The data type specifications in the exemplary “order.dtd” (Ex. E) include at least one logical structure. Similarly, slides 25-29 (Ex. I) depict using logical structure

building blocks to construct documents such as a purchase order (po.dtd). In general, a data type definition file (dtd) will include data type specifications for at least on logical data structure.

Claim 3: The data type specifications for the country field of the exemplary “order.dtd” (Ex. E) include at least one data structure mapping predefined sets of storage units for a particular logical structure in the definitions to respective entries in a list. The country code field is supported by a list in “codes.mod” (the codes module), which is incorporated into addresso.mod which, in turn, is part of order.dtd. With a bit of tracing from imdesc.xml, one can see where the reusable lists were part of the system. Use of country codes in addresses is a feature of CBL that extended the XML recommendation – the XML recommendation uses codes for language identification and not for addresses. See, W3C Recommendation, Extensible Markup Language (XML) 1.0, § 2.12 (Feb. 10, 1998) accessed at <http://www.w3.org/TR/1998/REC-xml-19980210#dt-app>. Turning to the July 25, 1998 presentation, slides 27-28 similarly illustrate use of country codes in a purchase order, defined by the file “po.dtd”.

Claim 4: In Exhibit D, the ENTITY “command” and the related ELEMENTs support registering definitions of documents and services in a repository in memory accessible to at least one node. The definitions of documents and services include logical structures and interpretation information for logical structures. In Exhibit I, Slide 23 depicts a service registration document in a registry accessible to discovery nodes on a network. Slide 31 indicates that CBL was already in use for demonstration applications, Project Seitai and GSA catalog interoperability. We found it very useful to post our interface definitions for programmers to rely on during development, such as the imdesc.xml interface definition.

Claim 5: Looking through files from the ingram/01 directory, which are reproduced in Exhibit H, one finds the data type definition corresponding to the sample interface definition imdesc.xml. Beginning with imdesc.xml on page 32 of 66, one sees reference to “imarkdsc.dtd” in the DOCTYPE statement of line six. This data type definition

appears on page 29 of 66, and incorporates by reference the “isrvprim.mod” module that appears on page 54 of 66. The isrvprim.mod defines elements of imdesc.xml, including service.name, service.location.pointer, and service.function. Thus, the sample “imdesc.xml” machine readable specification of Exhibit G complies with the imarkdsc.dtd definition of an interface document including logical structures for storing identifiers (e.g., service.name and/or input doctype) and references to definitions (dtd’s) of input and output documents. Similarly, in Exhibit I, slide 30, identifiers of particular transactions appear as “Submit Order” and “Track Order.” The references to definitions of the input and output documents are dtd file names. The “Loose Coupling” via Shared Document Definitions slide 22 makes it clear that the definition in slide 30 is a document that complies with a dtd, as in the ingram/01 files.

Claim 6: Both the “imdesc.xml” Exhibit G and the Slide 30 in Exhibit I specify one or more transactions supported by the interface.

Claim 7: Both the “imdesc.xml” Exhibit G and the Slide 30 in Exhibit I include references to documents used in the particular transactions, specifying the document dtd’s by name instead of providing a copy of them.

Claims 8-12: The input and output documents in both versions are depicted as XML documents. Generally, XML documents compliant with the February 1998 recommendation can be parsed or unparsed data. XML documents may encode text characters and the text characters may provide a natural language word. XML documents generally include markup data to identify sets of storage units.

Claim 13: The files reprinted in Exhibits D, E and H from directories 072, 075 and ingram/01 store document types that can be used in a plurality of transactions. Commands identified above were used to register these document types in a repository. Slides 25-29 depict a library of document types stored in a repository. The dtd definitions for the documents used in the transactions are referenced by names. For instance, in “imdesc.xml” we see the “ack.dtd” as the output document for several service functions.

Claim 14: The collections of files from directories include “imarkprt.dtd” and “imarkdsc.dtd” for market participant information and market description information. These modules are called out in Slide 28 of Exhibit I by slightly different and more easily readable names, “markpart.dtd” and “markdesc.dtd”. Some of these document types include identifications of participant processes.

Claim 15: The dtd files referenced in “imdesc.xml” and Slide 30, e.g., order.dtd and ack.dtd, are recognizable as compliant with a standard Extensible Markup Language XML.

Claim 16: The file “imdesc.xml” and the code excerpt in Slide 30 are recognizable as compliant with a standard Extensible Markup Language XML.

Claim 61: In the course of creating “imdesc.xml” and Slide 30, the authors of those documents went through the process of defining a machine readable interface definition including an input and output document. The versions of the transaction interface definition data structure were provided to network nodes that requested the definition data structures.

Claim 62: The data type specifications in the exemplary “order.dtd” include at least one logical structure. Similarly, slides 25-29 depict using logical structure building blocks to construct documents such as a purchase order (po.dtd). In general, a data type definition file (dtd) will include data type specifications for at least on logical data structure.

Claim 63: The data type specifications for the country field of the exemplary “order.dtd” (Ex. E) include at least one data structure mapping predefined sets of storage units for a particular logical structure in the definitions to respective entries in a list. The country code field is supported by a list in “codes.mod” (the codes module), which is incorporated into addresso.mod which, in turn, is part of order.dtd. With a bit of tracing from imdesc.xml, one can see where the reusable lists were part of the system. Use of country codes in addresses is a feature of CBL that extended the XML recommendation –

the XML recommendation uses codes for language identification and not for addresses. See, W3C Recommendation, Extensible Markup Language (XML) 1.0, § 2.12 (Feb. 10, 1998) accessed at <http://www.w3.org/TR/1998/REC-xml-19980210#dt-app>. Turning to the July 25, 1998 presentation, slides 27-28 similarly illustrate use of country codes in a purchase order, defined by the file "po.dtd".

Claim 64: Slide 23 (Ex. I) depicts a service registration document in a registry accessible to discovery nodes on a network. Slide 31 indicates that CBL was already in use for demonstration applications, Project Seitai and GSA catalog interoperability. We found it very useful to post our interface definitions for programmers to rely on during development, such as the imdesc.xml interface definition. In the January 1998 and earlier materials (Exs. D, E and H), we have printed some libraries of logical structures that we were using then. The Slides indicate that the libraries of logical structures were still in use six months later, when the conference presentation was made in July 1998.

Claim 65: Looking through files from the ingram/01 directory (Ex. H) one finds how imdesc.xml has been defined. Beginning with imdesc.xml on page 32 of 66, one sees reference to "imarkdsc.dtd" in the DOCTYPE statement of line six. This data type definition appears on page 29 of 66, and incorporates by reference the "isrvprim.mod" module that appears on page 54 of 66. The isrvprim.mod defines elements of imdesc.xml, including service.name, service.location.pointer, and service.function. Thus, the sample "imdesc.xml" machine readable specification of Exhibit G complies with the imarkdsc.dtd definition of an interface document including logical structures for storing identifiers (e.g., service.name and/or input doctype) and references to definitions (dtd's) of input and output documents. Similarly, in Exhibit I, slide 30, identifiers of particular transactions appear as "Submit Order" and "Track Order." The references to definitions of the input and output documents are dtd file names. The "Loose Coupling" via Shared Document Definitions slide 22 makes it clear that the definition in slide 30 is a document that complies with a dtd, as in the ingram/01 files.

Claims 66-70: The input and output documents in both versions (code and presentation) are depicted as XML documents. Generally, XML documents compliant with the February 1998 recommendation can be parsed or unparsed data. XML documents may encode text characters and the text characters may provide a natural language word. XML documents generally include markup data to identify sets of storage units.

Claim 71: The dtd files referenced in "imdesc.xml" and Slide 30 are recognizable as compliant with a standard Extensible Markup Language XML.

Claim 72: Our development work in 1997-98 included applying parsers to input documents, generating one or more Java event signals in response to the logical structure of the input documents, and having event listener programs respond to the event signals.

20. I am informed and believe that Examiner Huynh has questioned whether versions of CBL prior to the publicly released version 1.1 worked. In fact, CBL worked for its intended purpose for many months and many versions prior to the September 1998 release of public version 1.1.

I declare under penalty of perjury of the laws of the United States of America that the foregoing is true and correct. I make this declaration with the understanding and knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 U.S.C. and that making willful false statements would jeopardize the validity of my application and any patents issuing thereon.

Executed this 1 th day of July, 2008 in Honolulu, Hawaii.

Kevin Hughes
Kevin Hughes